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

Volume 2c: Command Reference: Register Addresses

For the 2014-2015 Intel Atom™ Processors, Celeron™ Processors and Pentium™
Processors based on the "Cherry Trail/Braswell" Platform
(Cherryview/Braswell graphics)

October 2015, Revision 1.1
Creative Commons License

You are free to Share - to copy, distribute, display, and perform the work under the following conditions:

- Attribution. You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).
- No Derivative Works. You may not alter, transform, or build upon this work.

Notices and Disclaimers

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

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

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

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

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

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

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

Copyright © 2015, Intel Corporation. All rights reserved.
Table of Contents

Advanced Features Length and Capabilities................................................................. 1
Advanced Scheduler Reset Request Messages............................................................ 3
Aggregate_Perf_Counter_A31.................................................................................. 5
Aggregate_Perf_Counter_A32.................................................................................. 6
Aggregate_Perf_Counter_A33.................................................................................. 7
Aggregate_Perf_Counter_A34.................................................................................. 8
Aggregate_Perf_Counter_A35.................................................................................. 9
Aggregate Perf Counter A1....................................................................................... 10
Aggregate Perf Counter A5....................................................................................... 11
Aggregate Perf Counter A7....................................................................................... 12
Aggregate Perf Counter A8....................................................................................... 13
Aggregate Perf Counter A9....................................................................................... 14
Aggregate Perf Counter A10..................................................................................... 15
Aggregate Perf Counter A11..................................................................................... 16
Aggregate Perf Counter A12..................................................................................... 17
Aggregate Perf Counter A13..................................................................................... 18
Aggregate Perf Counter A14..................................................................................... 19
Aggregate Perf Counter A15..................................................................................... 20
Aggregate Perf Counter A16..................................................................................... 21
Aggregate Perf Counter A17..................................................................................... 22
Aggregate Perf Counter A18..................................................................................... 23
Aggregate Perf Counter A21..................................................................................... 24
Aggregate Perf Counter A22..................................................................................... 25
Aggregate Perf Counter A23..................................................................................... 26
Aggregate Perf Counter A24..................................................................................... 27
Aggregate Perf Counter A25..................................................................................... 28
Aggregate Perf Counter A26..................................................................................... 29
Aggregate Perf Counter A27..................................................................................... 30
Aggregate Perf Counter A28..................................................................................... 31
Aggregate Perf Counter A29..................................................................................... 32
Aggregate Perf Counter A30..................................................................................... 33
AFCTLSTS ........................................................................................................... 34
<table>
<thead>
<tr>
<th>Register Name</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>All Engine Fault Register</td>
<td>35</td>
</tr>
<tr>
<td>ARAT CUTRIG HI</td>
<td>36</td>
</tr>
<tr>
<td>ARAT CUTRIG LO</td>
<td>37</td>
</tr>
<tr>
<td>ARAT Delta (LSB)</td>
<td>38</td>
</tr>
<tr>
<td>ARAT Delta (MSB)</td>
<td>39</td>
</tr>
<tr>
<td>ARAT POST CU BUSY</td>
<td>40</td>
</tr>
<tr>
<td>ARAT PRE CU BUSY</td>
<td>41</td>
</tr>
<tr>
<td>Arbiter Control Register</td>
<td>42</td>
</tr>
<tr>
<td>Arbiter Mode Control Register</td>
<td>44</td>
</tr>
<tr>
<td>ASLS</td>
<td>47</td>
</tr>
<tr>
<td>ASYNC_SLICE_COUNT</td>
<td>48</td>
</tr>
<tr>
<td>Auto Draw End Offset</td>
<td>49</td>
</tr>
<tr>
<td>Base of Data Stolen Memory</td>
<td>50</td>
</tr>
<tr>
<td>Base of GTT Stolen Memory</td>
<td>51</td>
</tr>
<tr>
<td>Batch Address Difference Register</td>
<td>52</td>
</tr>
<tr>
<td>Batch Buffer Head Pointer Register</td>
<td>53</td>
</tr>
<tr>
<td>Batch Buffer Per Context Pointer</td>
<td>55</td>
</tr>
<tr>
<td>Batch Buffer Start Head Pointer Register</td>
<td>58</td>
</tr>
<tr>
<td>Batch Buffer Start Head Pointer Register for Upper DWord</td>
<td>60</td>
</tr>
<tr>
<td>Batch Buffer State Register</td>
<td>61</td>
</tr>
<tr>
<td>Batch Buffer Upper Head Pointer Preemption Register</td>
<td>64</td>
</tr>
<tr>
<td>Batch Buffer Upper Head Pointer Register</td>
<td>65</td>
</tr>
<tr>
<td>BCS_PREEMPTION_HINT</td>
<td>66</td>
</tr>
<tr>
<td>BCS_PREEMPTION_HINT_UDW</td>
<td>68</td>
</tr>
<tr>
<td>BCS Active Upper Head Pointer Register</td>
<td>69</td>
</tr>
<tr>
<td>BCS Context ID Preemption Hint</td>
<td>70</td>
</tr>
<tr>
<td>BCS Context Sizes</td>
<td>71</td>
</tr>
<tr>
<td>BCS Context Timestamp Count</td>
<td>72</td>
</tr>
<tr>
<td>BCS Counter for the Blitter Engine</td>
<td>73</td>
</tr>
<tr>
<td>BCS Error Identity Register</td>
<td>74</td>
</tr>
<tr>
<td>BCS Error Mask Register</td>
<td>75</td>
</tr>
<tr>
<td>BCS Error Status Register</td>
<td>76</td>
</tr>
<tr>
<td>BCS Execute Condition Code Register</td>
<td>77</td>
</tr>
</tbody>
</table>
BCS General Purpose Register ................................................................. 79
BCS Hardware Status Mask Register .......................................................... 80
BCS IDLE Max Count.................................................................................. 81
BCS Idle Switch Delay................................................................................ 82
BCS Instruction Parser Mode Register .......................................................... 83
BCS Interrupt Mask Register .................................................................... 84
BCS Mode Register for Software Interface .................................................. 85
BCS PPGTT Directory Cacheline Valid Register .............................................. 87
BCS Primary DMA Engine Fetch Upper Address ........................................... 88
BCS Reported Timestamp Count.................................................................. 89
BCS Reset Control Register ...................................................................... 90
BCS Ring Buffer Next Context ID Register .................................................. 91
BCS Semaphore Polling Interval on Wait....................................................... 92
BCS SW Control......................................................................................... 93
BCS Watchdog Counter Threshold............................................................... 94
Bitstream Output Bit Count for the last Syntax Element Report Register ..... 95
Bitstream Output Byte Count Per Slice Report Register ................................. 96
Bitstream Output Minimal Size Padding Count Report Register ................. 97
Blitter Mode Register.................................................................................. 98
Blitter TLB Control Register ..................................................................... 101
BLT Context Element Descriptor (High Part) ............................................... 102
BLT Context Element Descriptor (Low Part) ................................................ 103
BLT Context Element Descriptor (Low Part) ................................................ 104
BLT PDP0/PML4/PASID Descriptor (High Part) ............................................. 105
BLT PDP0/PML4/PASID Descriptor (Low Part) .............................................. 106
BLT PDP1 Descriptor Register (High Part) .................................................... 107
BLT PDP1 Descriptor Register (Low Part) ..................................................... 108
BLT PDP2 Descriptor Register (High Part) .................................................... 109
BLT PDP2 Descriptor Register (Low Part) ..................................................... 110
BLT PDP3 Descriptor Register (High Part) .................................................... 111
BLT PDP3 Descriptor Register (Low Part) ..................................................... 112
Boolean.Counter_B0................................................................................... 113
Boolean.Counter_B1................................................................................... 114
Command Reference: Registers

Context Save Request To TDL................................................................. 177
Context Sizes ...................................................................................... 178
Context Status Buffer Contents ............................................................ 179
Control Register for Power Management .............................................. 181
Count Active Channels Dispatched ......................................................... 185
CS Context Timestamp Count ................................................................. 186
CS General Purpose Register ................................................................. 187
CS Power Management FSM ................................................................. 190
CSPREEMPT ......................................................................................... 194
CS Reset Control Register .................................................................... 195
Current Context Register ...................................................................... 196
Current Idle/Busy/Avg Count for Freq Down Recommendation ............. 198
Current Idle/Busy/Avg Count for Freq Up Recommendation ................ 199
Current Time in DOWN EI .................................................................... 200
Current Time in UP EI .......................................................................... 201
Customizable Event Creation 0-0 .......................................................... 202
Customizable Event Creation 1-0 .......................................................... 204
Customizable Event Creation 2-0 .......................................................... 206
Customizable Event Creation 3-0 .......................................................... 208
Customizable Event Creation 4-0 .......................................................... 210
Customizable Event Creation 5-0 .......................................................... 212
Customizable Event Creation 6-0 .......................................................... 214
Customizable Event Creation 7-0 .......................................................... 216
CVS TLB LRA 0 ..................................................................................... 218
CVS TLB LRA 1 ..................................................................................... 220
CVS TLB LRA 2 ..................................................................................... 222
Depth/Early Depth TLB Partitioning Register .......................................... 223
DID ......................................................................................................... 224
DS Invocation Counter .......................................................................... 226
DX9 Constants Not Consumed By RCS ............................................... 227
DX9 Constants Psed By RCS ................................................................. 228
ECO Bits - Bus Reset Domain with lock bit ......................................... 229
ECO Bits - Device Reset Domain .......................................................... 231
EU Metrics for Event10 Low ................................................................. 274
EU Metrics for Event11 High ................................................................. 275
EU Metrics for Event11 Low ................................................................. 276
EU Metrics for Event12 High ................................................................. 277
EU Metrics for Event12 Low ................................................................. 278
Event selection and base counters ......................................................... 279
Event Selection and Base Counters1 ....................................................... 282
Execlist Status ....................................................................................... 283
Execlist Submit Port Register ................................................................. 286
Execute Condition Code Register .......................................................... 289
FAULT_TLB_RD_DATA0 Register ............................................................ 291
FAULT_TLB_RD_DATA1 Register ............................................................ 292
Fault Switch Out .................................................................................... 293
FBC_RT_BASE_ADDR_REGISTER .......................................................... 294
FBC_RT_BASE_ADDR_REGISTER_UPPER ............................................. 296
FD .......................................................................................................... 297
FF Performance ...................................................................................... 298
First Buffer Size and Start .................................................................... 300
Flexible EU Event Control 0 ................................................................. 302
Flexible EU Event Control 1 ................................................................. 304
Flexible EU Event Control 2 ................................................................. 306
Flexible EU Event Control 3 ................................................................. 308
Flexible EU Event Control 4 ................................................................. 310
Flexible EU Event Control 5 ................................................................. 312
Flexible EU Event Control 6 ................................................................. 314
FORCE_TO_NONPRIV .......................................................................... 316
Frame count and Draw call number ....................................................... 322
FUSEWORD0 ........................................................................................ 323
FUSEWORD1 ........................................................................................ 324
FUSEWORD2 ........................................................................................ 325
FUSEWORD3 ........................................................................................ 326
FUSEWORD4 ........................................................................................ 327
FUSEWORD5 ........................................................................................ 328
Command Reference: Registers

GAC_GAM WR Arbitration Register 3 ................................................................. 363
GAM and SA Communication Register .......................................................... 364
Gam Fub Done1 Lookup Register .............................................................. 367
Gam Fub Done Lookup Register .............................................................. 368
GAM Put Delay ......................................................................................... 370
GAMT_DONE Register .............................................................................. 371
GAMT_ECO_REG_RO_IA ............................................................................ 373
GAMT_ECO_REG_RW_IA ............................................................................ 374
GAMT Arbiter Mode Control ....................................................................... 375
GAMW_ECO_BUS_RO_IA ............................................................................ 378
GAMW_ECO_BUS_RW_IA ............................................................................ 379
GAMW_ECO_DEV_RO_IA ............................................................................ 380
GAMW_ECO_DEV_RW_IA ............................................................................ 381
GAMW Power Context Save ........................................................................ 382
GARB Messaging Register for Boot Controller .............................................. 384
GARB Messaging Register for Clocking Unit ................................................ 386
Gather Constants Not Consumed By RCS .............................................. 387
GDR Per Client Write Drop Enables .......................................................... 388
GDR Write Drop ....................................................................................... 390
General Purpose Power Management Performance Idle Hysteresis .............. 391
GFX_FLSH_CNT ....................................................................................... 392
GFX Arbiter Client Priority Control .............................................................. 393
GFX Context Element Descriptor (High Part) ............................................... 395
GFX Context Element Descriptor (Low Part) ............................................... 396
GFX Context Element Descriptor (Low Part) ............................................... 398
GFX PDP0/PML4/PASID Descriptor (High Part) .............................................. 400
GFX PDP0/PML4/PASID Descriptor (Low Part) .............................................. 401
GFX PDP1 Descriptor Register (High Part) .............................................. 402
GFX PDP1 Descriptor Register (Low Part) .............................................. 403
GFX PDP2 Descriptor Register (High Part) .............................................. 404
GFX PDP2 Descriptor Register (Low Part) .............................................. 405
GFX PDP3 Descriptor Register (High Part) .............................................. 406
GFX PDP3 Descriptor Register (Low Part) .............................................. 407
GT INTERRUPT1 ENABLE REGISTER ................................................................. 408
Global Clear Value Register 0 ........................................................................ 411
Global Clear Value Register 1 ........................................................................ 412
Global Clear Value Register 2 ........................................................................ 413
Global Clear Value Register 3 ........................................................................ 414
Global Invalidation Register ......................................................................... 415
Global System Interrupt Routine .................................................................. 416
GMADR_LSB ................................................................................................. 417
GMADR_MSB ................................................................................................. 419
GMBC Message Register ............................................................................... 420
GO Messaging Register for GAMunit .............................................................. 422
Go Protocol GAM Request ........................................................................... 424
GPA to HPA Translation Request .................................................................. 428
GPA value for GPA to HPA Translation ......................................................... 431
GPGPU Context Restore Request To TDL ...................................................... 432
GPGPU Context Save Request To TDL ........................................................... 434
GPGPU Dispatch Dimension X ...................................................................... 435
GPGPU Dispatch Dimension Y ...................................................................... 436
GPGPU Dispatch Dimension Z ...................................................................... 437
GPU_Ticks_Counter ...................................................................................... 438
Graphics Device Reset Control ..................................................................... 439
Graphics Mode Register ............................................................................... 441
GS Domain Clock Gate Control Register ...................................................... 445
GS Invocation Counter .................................................................................. 448
GS Primitives Counter .................................................................................. 449
GT_CR_POWER_METER_CTRL ..................................................................... 450
GTFIFOCTL ................................................................................................. 452
GT Function Level Reset Control Message .................................................... 454
GT INTERRUPT 0 ENABLE REGISTER ......................................................... 455
GT INTERRUPT 0 IDENTITY REGISTER ........................................................ 458
GT INTERRUPT 0 MASK REGISTER ............................................................... 461
GT INTERRUPT 0 STATUS REGISTER ............................................................ 465
GT INTERRUPT1 ENABLE REGISTER ........................................................... 468
Command Reference: Registers

GT INTERRUPT1 IDENTITY REGISTER ................................................................. 471
GT INTERRUPT1 MASK REGISTER ................................................................. 474
GT INTERRUPT1 STATUS REGISTER .............................................................. 477
GT INTERRUPT3 ENABLE REGISTER ............................................................ 480
GT INTERRUPT3 IDENTITY REGISTER .............................................................. 482
GT INTERRUPT3 MASK REGISTER ................................................................. 484
GT INTERRUPT3 STATUS REGISTER ............................................................... 486
GTL_PW_STAT .................................................................................................. 488
GTL_SURVIVE .................................................................................................. 490
GTL_WAKE ......................................................................................................... 492
GT Mode Register .......................................................................................... 494
GTSCRATCH1 ..................................................................................................... 497
GTSCRATCH2 ..................................................................................................... 498
GTSCRATCH3 ..................................................................................................... 499
GTSCRATCH4 ..................................................................................................... 500
GTSCRATCH5 ..................................................................................................... 501
GTSCRATCH6 ..................................................................................................... 502
GTSCRATCH7 ..................................................................................................... 503
GTSCRATCH8 ..................................................................................................... 504
GTSCRATCHPAD0 ............................................................................................... 505
GTSCRATCHPAD1 ............................................................................................... 506
GTSCRATCHPAD2 ............................................................................................... 507
GTSCRATCHPAD3 ............................................................................................... 508
GTSCRATCHPAD4 ............................................................................................... 509
GTSCRATCHPAD5 ............................................................................................... 510
GTSCRATCHPAD6 ............................................................................................... 511
GTSCRATCHPAD7 ............................................................................................... 512
GTT Cache Enable ........................................................................................... 513
GTTMMADR_LSB ............................................................................................... 514
GTTMMADR_MSB ............................................................................................... 516
GU_CTL0 ........................................................................................................... 517
Hardware Status Mask Register ....................................................................... 520
Hardware Status Page Address Register .......................................................... 522
IA32 MTRR PHYSBASE1 Low ......................................................... 558
IA32 MTRR PHYSBASE2 High ....................................................... 559
IA32 MTRR PHYSBASE2 Low ....................................................... 560
IA32 MTRR PHYSBASE3 High ....................................................... 561
IA32 MTRR PHYSBASE3 Low ....................................................... 562
IA32 MTRR PHYSBASE4 High ....................................................... 563
IA32 MTRR PHYSBASE4 Low ....................................................... 564
IA32 MTRR PHYSBASE5 High ....................................................... 565
IA32 MTRR PHYSBASE5 Low ....................................................... 566
IA32 MTRR PHYSBASE6 High ....................................................... 567
IA32 MTRR PHYSBASE6 Low ....................................................... 568
IA32 MTRR PHYSBASE7 High ....................................................... 569
IA32 MTRR PHYSBASE7 Low ....................................................... 570
IA32 MTRR PHYSBASE8 High ....................................................... 571
IA32 MTRR PHYSBASE8 Low ....................................................... 572
IA32 MTRR PHYSBASE9 High ....................................................... 573
IA32 MTRR PHYSBASE9 Low ....................................................... 574
IA32 MTRR PHYSMASK0 High ..................................................... 575
IA32 MTRR PHYSMASK0 Low ..................................................... 576
IA32 MTRR PHYSMASK1 High ..................................................... 577
IA32 MTRR PHYSMASK1 Low ..................................................... 578
IA32 MTRR PHYSMASK2 High ..................................................... 579
IA32 MTRR PHYSMASK2 Low ..................................................... 580
IA32 MTRR PHYSMASK3 High ..................................................... 581
IA32 MTRR PHYSMASK3 Low ..................................................... 582
IA32 MTRR PHYSMASK4 High ..................................................... 583
IA32 MTRR PHYSMASK4 Low ..................................................... 584
IA32 MTRR PHYSMASK5 High ..................................................... 585
IA32 MTRR PHYSMASK5 Low ..................................................... 586
IA32 MTRR PHYSMASK6 High ..................................................... 587
IA32 MTRR PHYSMASK6 Low ..................................................... 588
IA32 MTRR PHYSMASK7 High ..................................................... 589
IA32 MTRR PHYSMASK7 Low ..................................................... 590
command reference: registers

L3 SQC register 4 ................................................................. 640
L3 SQC registers 1 ............................................................. 643
L3 SQC registers 2 ............................................................. 647
L3 SQC registers 3 ............................................................. 651
LBCF config save msg ......................................................... 655
LBCF DPF Error log register 0 .............................................. 656
LBCF DPF Error log register 1 .............................................. 657
LBCF DPF Error log register 2 .............................................. 658
LBCF DPF Error log register 3 .............................................. 659
LBCF DPF Error log register 4 .............................................. 660
LBCF DPF Error log register 5 .............................................. 661
LBCF DPF Error log register 6 .............................................. 662
LBCF DPF Error log register 7 .............................................. 663
LBCF DPF Error log register 8 .............................................. 664
LBCF DPF Error log register 9 .............................................. 665
LBCF DPF Error log register 10 ............................................ 666
LBCF DPF Error log register 11 ............................................ 667
LBCF DPF Error log register 12 ............................................ 668
LBCF DPF Error log register 13 ............................................ 669
LBCF DPF Error log register 14 ............................................ 670
LBCF DPF Error log register 15 ............................................ 671
LBCF DPF Error log register 16 ............................................ 672
LBCF DPF Error log register 17 ............................................ 673
LBCF DPF Error log register 18 ............................................ 674
LBCF DPF Error log register 19 ............................................ 675
LBCF DPF Error log register 20 ............................................ 676
LBCF DPF Error log register 21 ............................................ 677
LBCF DPF Error log register 22 ............................................ 678
LBCF DPF Error log register 23 ............................................ 679
LBCF DPF Error log register 24 ............................................ 680
LBCF DPF Error log register 25 ............................................ 681
LBCF DPF Error log register 26 ............................................ 682
LBCF DPF Error log register 27 ............................................ 683
LBCF DPF Error log register 28 ................................................................. 684
LBCF DPF Error log register 29 ................................................................. 685
LBCF DPF Error log register 30 ................................................................. 686
LBCF DPF Error log register 31 ................................................................. 687
LBCF DPF Error log register 32 ................................................................. 688
LBCF DPF Error log register 33 ................................................................. 689
LBCF DPF Error log register 34 ................................................................. 690
LBCF DPF Error log register 35 ................................................................. 691
LBCF DPF Error log register 36 ................................................................. 692
LBCF DPF Error log register 37 ................................................................. 693
LBCF DPF Error log register 38 ................................................................. 694
LBCF DPF Error log register 39 ................................................................. 695
LBCF DPF Error log register 40 ................................................................. 696
LBCF DPF Error log register 41 ................................................................. 697
LBCF DPF Error log register 42 ................................................................. 698
LBCF DPF Error log register 43 ................................................................. 699
LBCF DPF Error log register 44 ................................................................. 700
LBCF DPF Error log register 45 ................................................................. 701
LBCF DPF Error log register 46 ................................................................. 702
LBCF DPF Error log register 47 ................................................................. 703
LBS config bits ....................................................................................... 704
LEAKAGECOUNTER ................................................................................. 705
LEAKAGECOUNTERCTL ....................................................................... 706
LEAKAGEWEIGHT1 .............................................................................. 707
LEAKAGEWEIGHT2 .............................................................................. 708
LNCF config save msg .......................................................................... 709
Load Indirect Base Vertex ...................................................................... 710
Load Indirect Instance Count ................................................................. 711
Load Indirect Start Instance .................................................................. 712
Load Indirect Start Vertex .................................................................... 713
Load Indirect Vertex Count .................................................................. 714
LOW 2X FREQUENCY THRESHOLD ....................................................... 715
LPFC control register ........................................................................... 716
<table>
<thead>
<tr>
<th>Register Name</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>LTCD Error Injection Register</td>
<td>717</td>
</tr>
<tr>
<td>MA</td>
<td>720</td>
</tr>
<tr>
<td>Main Graphic Arbiter Error Report</td>
<td>721</td>
</tr>
<tr>
<td>Main Graphic Arbiter Error Report 2</td>
<td>725</td>
</tr>
<tr>
<td>Main Graphic Arbiter Error Report 3</td>
<td>726</td>
</tr>
<tr>
<td>Main Graphic Arbiter Error Report Register</td>
<td>729</td>
</tr>
<tr>
<td>MASTER_INT_CTL</td>
<td>730</td>
</tr>
<tr>
<td>Master start timer</td>
<td>732</td>
</tr>
<tr>
<td>Max Outstanding Pending TLB Requests 0</td>
<td>733</td>
</tr>
<tr>
<td>Max Outstanding Pending TLB Requests 1</td>
<td>735</td>
</tr>
<tr>
<td>MAX Requests Allowed - GAM</td>
<td>737</td>
</tr>
<tr>
<td>MAX Requests Allowed - MFX</td>
<td>739</td>
</tr>
<tr>
<td>MAX Requests Allowed - VEBX and BLT</td>
<td>741</td>
</tr>
<tr>
<td>MBC Control Register</td>
<td>742</td>
</tr>
<tr>
<td>MBDSM</td>
<td>744</td>
</tr>
<tr>
<td>MBGSM</td>
<td>745</td>
</tr>
<tr>
<td>MD</td>
<td>746</td>
</tr>
<tr>
<td>MEDFW_ACK</td>
<td>747</td>
</tr>
<tr>
<td>Media 1 TLB Control Register</td>
<td>748</td>
</tr>
<tr>
<td>Media 2 TLB Control Register</td>
<td>749</td>
</tr>
<tr>
<td>Media forcewake request</td>
<td>750</td>
</tr>
<tr>
<td>Media Power Meter Counter</td>
<td>753</td>
</tr>
<tr>
<td>Media Power Meter Counter No Clear</td>
<td>754</td>
</tr>
<tr>
<td>MEDIARC0COUNTER</td>
<td>755</td>
</tr>
<tr>
<td>MEDIARC1COUNTER</td>
<td>756</td>
</tr>
<tr>
<td>MEDIARC6COUNTER</td>
<td>757</td>
</tr>
<tr>
<td>MEMBOUNDCOUNTER</td>
<td>758</td>
</tr>
<tr>
<td>Message Register</td>
<td>759</td>
</tr>
<tr>
<td>Messaging Register for GPMunit</td>
<td>762</td>
</tr>
<tr>
<td>Messaging Register for MDRBunit</td>
<td>765</td>
</tr>
<tr>
<td>Messaging Register for MGSRunit</td>
<td>766</td>
</tr>
<tr>
<td>MFC_AVC_CABAC_INSERTION_COUNT</td>
<td>767</td>
</tr>
<tr>
<td>MFC_AVC Bitstream Decoding Front-End Parsing Logic Error Counter</td>
<td>768</td>
</tr>
<tr>
<td>Register Description</td>
<td>Page Number</td>
</tr>
<tr>
<td>---------------------------------------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>MFC Image Status Control</td>
<td>769</td>
</tr>
<tr>
<td>MFC Image Status Mask</td>
<td>770</td>
</tr>
<tr>
<td>MFC QP Status Count</td>
<td>771</td>
</tr>
<tr>
<td>MFD Error Status</td>
<td>772</td>
</tr>
<tr>
<td>MFD Picture Parameter</td>
<td>773</td>
</tr>
<tr>
<td>MFX_Memory_Latency_Count1</td>
<td>774</td>
</tr>
<tr>
<td>MFX0 Context Element Descriptor (High Part)</td>
<td>775</td>
</tr>
<tr>
<td>MFX0 Context Element Descriptor (Low Part)</td>
<td>776</td>
</tr>
<tr>
<td>MFX0 Context Element Descriptor (Low Part)</td>
<td>777</td>
</tr>
<tr>
<td>MFX0 PDP0/PML4/PASID Descriptor (High Part)</td>
<td>778</td>
</tr>
<tr>
<td>MFX0 PDP0/PML4/PASID Descriptor (Low Part)</td>
<td>779</td>
</tr>
<tr>
<td>MFX0 PDP1 Descriptor Register (High Part)</td>
<td>780</td>
</tr>
<tr>
<td>MFX0 PDP1 Descriptor Register (Low Part)</td>
<td>781</td>
</tr>
<tr>
<td>MFX0 PDP2 Descriptor Register (High Part)</td>
<td>782</td>
</tr>
<tr>
<td>MFX0 PDP2 Descriptor Register (Low Part)</td>
<td>783</td>
</tr>
<tr>
<td>MFX0 PDP3 Descriptor Register (High Part)</td>
<td>784</td>
</tr>
<tr>
<td>MFX0 PDP3 Descriptor Register (Low Part)</td>
<td>785</td>
</tr>
<tr>
<td>MFX1 Context Element Descriptor (High Part)</td>
<td>786</td>
</tr>
<tr>
<td>MFX1 Context Element Descriptor (Low Part)</td>
<td>787</td>
</tr>
<tr>
<td>MFX1 Context Element Descriptor (Low Part)</td>
<td>788</td>
</tr>
<tr>
<td>MFX1 PDP0/PML4/PASID Descriptor (High Part)</td>
<td>789</td>
</tr>
<tr>
<td>MFX1 PDP0/PML4/PASID Descriptor (Low Part)</td>
<td>790</td>
</tr>
<tr>
<td>MFX1 PDP1 Descriptor Register (High Part)</td>
<td>791</td>
</tr>
<tr>
<td>MFX1 PDP1 Descriptor Register (Low Part)</td>
<td>792</td>
</tr>
<tr>
<td>MFX1 PDP2 Descriptor Register (High Part)</td>
<td>793</td>
</tr>
<tr>
<td>MFX1 PDP2 Descriptor Register (Low Part)</td>
<td>794</td>
</tr>
<tr>
<td>MFX1 PDP3 Descriptor Register (High Part)</td>
<td>795</td>
</tr>
<tr>
<td>MFX1 PDP3 Descriptor Register (Low Part)</td>
<td>796</td>
</tr>
<tr>
<td>MFX Frame BitStream SE/BIN Count</td>
<td>797</td>
</tr>
<tr>
<td>MFX Frame Macroblock Count</td>
<td>798</td>
</tr>
<tr>
<td>MFX Frame Motion Comp Miss Count</td>
<td>799</td>
</tr>
<tr>
<td>MFX Frame Motion Comp Read Count</td>
<td>800</td>
</tr>
<tr>
<td>MFX Frame Row-Stored/BitStream Read Count</td>
<td>801</td>
</tr>
</tbody>
</table>
# Command Reference: Registers

<table>
<thead>
<tr>
<th>Register Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>MFX LRA 0</td>
<td>802</td>
</tr>
<tr>
<td>MFX LRA 1</td>
<td>803</td>
</tr>
<tr>
<td>MFX LRA 2</td>
<td>804</td>
</tr>
<tr>
<td>MFX LRA SL1 0</td>
<td>805</td>
</tr>
<tr>
<td>MFX LRA SL1 1</td>
<td>806</td>
</tr>
<tr>
<td>MFX LRA SL1 2</td>
<td>807</td>
</tr>
<tr>
<td>MFX Memory Latency Count2</td>
<td>808</td>
</tr>
<tr>
<td>MFX Memory Latency Count3</td>
<td>809</td>
</tr>
<tr>
<td>MFX Memory Latency Count4</td>
<td>810</td>
</tr>
<tr>
<td>MFX Pipeline Status Flags</td>
<td>811</td>
</tr>
<tr>
<td>MFX Slice Performance Count</td>
<td>813</td>
</tr>
<tr>
<td>MGGC</td>
<td>814</td>
</tr>
<tr>
<td>MGSR2GAM Message Register</td>
<td>817</td>
</tr>
<tr>
<td>Misc. Clocking / Reset Control Registers</td>
<td>820</td>
</tr>
<tr>
<td>MISC. CTX control register</td>
<td>822</td>
</tr>
<tr>
<td>Misc. Reset Control Register</td>
<td>823</td>
</tr>
<tr>
<td>Miscellaneous Message Register for Power Management Unit</td>
<td>824</td>
</tr>
<tr>
<td>MISR0</td>
<td>825</td>
</tr>
<tr>
<td>MISR1</td>
<td>826</td>
</tr>
<tr>
<td>MISR2</td>
<td>827</td>
</tr>
<tr>
<td>MISR3</td>
<td>828</td>
</tr>
<tr>
<td>MISR4</td>
<td>829</td>
</tr>
<tr>
<td>MMIO_INDEX</td>
<td>830</td>
</tr>
<tr>
<td>Mode Register for GAB</td>
<td>831</td>
</tr>
<tr>
<td>Mode Register for GAC</td>
<td>832</td>
</tr>
<tr>
<td>Mode Register for GAFS</td>
<td>833</td>
</tr>
<tr>
<td>MSAC</td>
<td>834</td>
</tr>
<tr>
<td>MSI_CAPID_MC</td>
<td>837</td>
</tr>
<tr>
<td>MTRR Capability Register 0</td>
<td>839</td>
</tr>
<tr>
<td>MTRR Capability Register 1</td>
<td>840</td>
</tr>
<tr>
<td>MTRR Default Type Register 0</td>
<td>841</td>
</tr>
<tr>
<td>MTRR Default Type Register 1</td>
<td>842</td>
</tr>
<tr>
<td>MT Virtual Page Address Registers</td>
<td>843</td>
</tr>
</tbody>
</table>
POWER_WELL_SS1_SIG2 .................................................................................................................. 893
Power Context Save ...................................................................................................................... 895
Power context Save Register for LPFC .......................................................................................... 897
Power Context Save request ......................................................................................................... 898
POWERDOWN_STATE .................................................................................................................. 899
POWERDOWN_WAIT1 .................................................................................................................... 901
POWERDOWN_WAIT2 .................................................................................................................... 902
Power Down/Up Control ................................................................................................................ 903
Power Enable stagger control ....................................................................................................... 904
Power Meter Weight for gti_idle_cz ............................................................................................. 905
Power Meter Weight for gti_idle_gs ............................................................................................. 906
Power Meter Weight for media/render ....................................................................................... 907
POWERUP_STATE ........................................................................................................................ 908
POWERUP_WAIT1 .......................................................................................................................... 910
POWERUP_WAIT2 .......................................................................................................................... 911
PPGTT Page Fault Data Registers ............................................................................................... 912
Predicate Rendering Data Result .................................................................................................. 913
Predicate Rendering Data Result 1 ............................................................................................... 914
Predicate Rendering Data Result 2 ............................................................................................... 915
Predicate Rendering Data Storage ............................................................................................... 916
Predicate Rendering Temporary Register0 .................................................................................. 917
Predicate Rendering Temporary Register1 .................................................................................. 918
Previous Idle/Busy/Avg Count for Freq Down Recommendation ............................................... 919
Previous Idle/Busy/Avg Count for Freq Up Recommendation ..................................................... 920
Primitives Generated By VF ........................................................................................................ 921
Private PAT .................................................................................................................................... 922
Private PAT .................................................................................................................................... 923
PS Depth Count ............................................................................................................................. 924
PS Depth Count for Slice0 .............................................................................................................. 925
PS Depth Count for Slice1 .............................................................................................................. 926
PS Depth Count for Slice2 .............................................................................................................. 927
PS Depth Count for Slice3 .............................................................................................................. 928
PS Invocation Count ...................................................................................................................... 929
<table>
<thead>
<tr>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>PS Invocation Count for Slice0</td>
<td>930</td>
</tr>
<tr>
<td>PS Invocation Count for Slice1</td>
<td>931</td>
</tr>
<tr>
<td>PS Invocation Count for Slice2</td>
<td>932</td>
</tr>
<tr>
<td>PS Invocation Count for Slice3</td>
<td>933</td>
</tr>
<tr>
<td>PTBR_PAGE_POOL_OOM_EVENT_REGISTER</td>
<td>934</td>
</tr>
<tr>
<td>PTBR Page Pool Size on Out Of Memory</td>
<td>935</td>
</tr>
<tr>
<td>PTE SW Fault Repair High</td>
<td>936</td>
</tr>
<tr>
<td>PTE SW Fault Repair Low</td>
<td>937</td>
</tr>
<tr>
<td>Punit to Gunit Message</td>
<td>938</td>
</tr>
<tr>
<td>PWRCTXSAVE Message Register for Power Managment Unit</td>
<td>940</td>
</tr>
<tr>
<td>PWRMTR_WT1_EEVT1TO4</td>
<td>941</td>
</tr>
<tr>
<td>PWRMTR_WT1_EEVT5TO8</td>
<td>942</td>
</tr>
<tr>
<td>PWRMTR_WT1_EEVT9TO12</td>
<td>943</td>
</tr>
<tr>
<td>PWRMTR_WT1_EEVT13TO16</td>
<td>944</td>
</tr>
<tr>
<td>PWRMTR_WT1_EEVT17TO20</td>
<td>945</td>
</tr>
<tr>
<td>PWRMTR_WT1_EEVT21TO24</td>
<td>946</td>
</tr>
<tr>
<td>PWRMTR_WT1_EEVT25TO28</td>
<td>947</td>
</tr>
<tr>
<td>PWRMTR_WT1_EEVT29TO32</td>
<td>948</td>
</tr>
<tr>
<td>PWRMTR_WT1_EEVT33TO36</td>
<td>949</td>
</tr>
<tr>
<td>PWRMTR_WT1_EEVT37TO40</td>
<td>950</td>
</tr>
<tr>
<td>PWRMTR_WT1_EEVT41TO44</td>
<td>951</td>
</tr>
<tr>
<td>PWRMTR_WT1_EEVT45TO48</td>
<td>952</td>
</tr>
<tr>
<td>PWRMTR_WT1_EEVT49TO52</td>
<td>953</td>
</tr>
<tr>
<td>PWRMTR_WT1_EEVT53TO56</td>
<td>954</td>
</tr>
<tr>
<td>PWRMTR_WT1_EEVT57TO58</td>
<td>955</td>
</tr>
<tr>
<td>PWRMTR_WT1_MEVT1TO4</td>
<td>956</td>
</tr>
<tr>
<td>PWRMTR_WT1_MEVT5TO8</td>
<td>957</td>
</tr>
<tr>
<td>PWRMTR_WT1_MEVT9TO12</td>
<td>958</td>
</tr>
<tr>
<td>PWRMTR_WT1_MEVT13TO16</td>
<td>959</td>
</tr>
<tr>
<td>PWRMTR_WT1_MEVT17TO20</td>
<td>960</td>
</tr>
<tr>
<td>PWRMTR_WT1_MEVT21TO24</td>
<td>961</td>
</tr>
<tr>
<td>PWRMTR_WT1_MEVT25TO28</td>
<td>962</td>
</tr>
<tr>
<td>PWRMTR_WT1_MEVT29TO32</td>
<td>963</td>
</tr>
</tbody>
</table>
Command Reference: Registers

PWRMTR_WT1_MEVT33TO36.................................................................964
PWRMTR_WT1_REVT1TO4 .................................................................965
PWRMTR_WT1_REVT5TO8 .................................................................966
PWRMTR_WT1_REVT9TO12 .................................................................967
PWRMTR_WT1_REVT13TO16 ...............................................................968
PWRMTR_WT1_REVT17TO20 ..............................................................969
PWRMTR_WT1_REVT21TO24 ..............................................................970
PWRMTR_WT1_REVT25TO28 ..............................................................971
PWRMTR_WT1_REVT29TO32 ..............................................................972
PWRMTR_WT1_REVT33TO36 ..............................................................973
PWRMTR_WT1_REVT37TO40 ..............................................................974
PWRMTR_WT1_REVT41TO44 ..............................................................975
PWRMTR_WT1_REVT45TO48 ..............................................................976
PWRMTR_WT1_REVT49TO52 ..............................................................977
PWRMTR_WT1_REVT53TO56 ..............................................................978
PWRMTR_WT1_REVT57TO60 ..............................................................979
PWRMTR_WT1_REVT61TO64 ..............................................................980
PWRMTR_WT1_REVT65TO68 ..............................................................981
PWRMTR_WT1_REVT69TO70 ..............................................................982
PWRMTR_WT2_EEVT1TO4 .................................................................983
PWRMTR_WT2_EEVT5TO8 .................................................................984
PWRMTR_WT2_EEVT9TO12 ...............................................................985
PWRMTR_WT2_EEVT13TO16 .............................................................986
PWRMTR_WT2_EEVT17TO20 .............................................................987
PWRMTR_WT2_EEVT21TO24 .............................................................988
PWRMTR_WT2_EEVT25TO28 .............................................................989
PWRMTR_WT2_EEVT29TO32 .............................................................990
PWRMTR_WT2_EEVT33TO36 .............................................................991
PWRMTR_WT2_EEVT37TO40 .............................................................992
PWRMTR_WT2_EEVT41TO44 .............................................................993
PWRMTR_WT2_EEVT45TO48 .............................................................994
PWRMTR_WT2_EEVT49TO52 .............................................................995
PWRMTR_WT2_EEVT53TO56 .............................................................996
<table>
<thead>
<tr>
<th>Register Name</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>PWRMTREVTORE1</td>
<td>1030</td>
</tr>
<tr>
<td>PWRMTREVTM0</td>
<td>1031</td>
</tr>
<tr>
<td>PWRMTREVTM1</td>
<td>1032</td>
</tr>
<tr>
<td>PWRMTREVTOR0</td>
<td>1033</td>
</tr>
<tr>
<td>PWRMTREVTOR1</td>
<td>1034</td>
</tr>
<tr>
<td>PWRMTREVTOR2</td>
<td>1035</td>
</tr>
<tr>
<td>RAM Clock Gating Control 1</td>
<td>1036</td>
</tr>
<tr>
<td>RAM Clock Gating Control 1</td>
<td>1042</td>
</tr>
<tr>
<td>RAM Clock Gating Control 2</td>
<td>1048</td>
</tr>
<tr>
<td>RAM Clock Gating Control 2</td>
<td>1053</td>
</tr>
<tr>
<td>RAM Clock Gating Control 3</td>
<td>1058</td>
</tr>
<tr>
<td>RC6 Wake Rate Limit</td>
<td>1061</td>
</tr>
<tr>
<td>RCC LRA 0</td>
<td>1062</td>
</tr>
<tr>
<td>RCC LRA 1</td>
<td>1063</td>
</tr>
<tr>
<td>RCC Virtual page Address Registers</td>
<td>1064</td>
</tr>
<tr>
<td>RC Evaluation Interval</td>
<td>1065</td>
</tr>
<tr>
<td>RC Idle Hysteresis</td>
<td>1066</td>
</tr>
<tr>
<td>RCS_PREEMPTION_HINT</td>
<td>1067</td>
</tr>
<tr>
<td>RCS_PREEMPTION_HINT_UDW</td>
<td>1069</td>
</tr>
<tr>
<td>RCS Batch Buffer State Register</td>
<td>1070</td>
</tr>
<tr>
<td>RCS Context Preemption Hint</td>
<td>1071</td>
</tr>
<tr>
<td>RC Wake Counter</td>
<td>1072</td>
</tr>
<tr>
<td>RCZ Virtual Page Address Registers</td>
<td>1073</td>
</tr>
<tr>
<td>Ready Bit Vector 0 for TLBPEND registers</td>
<td>1074</td>
</tr>
<tr>
<td>Ready Bit Vector 1 for TLBPEND registers</td>
<td>1075</td>
</tr>
<tr>
<td>Render C State Control 1</td>
<td>1076</td>
</tr>
<tr>
<td>Render forcemailawate acknowledge</td>
<td>1077</td>
</tr>
<tr>
<td>Render forcewake request</td>
<td>1078</td>
</tr>
<tr>
<td>Render Geyserville Mode Control 1</td>
<td>1081</td>
</tr>
<tr>
<td>Render Mode Register for Software Interface</td>
<td>1084</td>
</tr>
<tr>
<td>Render Performance Status 1</td>
<td>1088</td>
</tr>
<tr>
<td>Render Performance Status Register</td>
<td>1089</td>
</tr>
<tr>
<td>Render Power Clock State Register</td>
<td>1090</td>
</tr>
</tbody>
</table>
Render Power Meter Counter .................................................................................................................1092
Render Power Meter Counter No Clear .................................................................................................1093
Render Promotion Timer - RC6 ..............................................................................................................1094
RENDERR0COUNTER ..............................................................................................................................1095
RENDERR1COUNTER ..............................................................................................................................1096
RENDERR6COUNTER ..............................................................................................................................1097
Render TLB Control Register ...............................................................................................................1098
Render Watchdog Counter ....................................................................................................................1099
Render Watchdog Counter Threshold .................................................................................................1100
Reported BitRateControl parameter Mask ..........................................................................................1101
Reported BitRateControl parameter Status .........................................................................................1103
Reported Bitstream Output Bit Count for Syntax Elements Only Register ........................................1105
Reported Bitstream Output Byte Count per Frame Register ..................................................................1106
Reported Bitstream Output CABAC Bin Count Register ......................................................................1107
Reported Final Bitstream Byte Count ...................................................................................................1108
Reported Frame Zero Padding Byte Count ...........................................................................................1109
Reported Timestamp Count ................................................................................................................1110
Reset Flow Control Messages .............................................................................................................1112
RESET Messaging Register for Clocking Unit ......................................................................................1115
Resource Streamer Context Offset ........................................................................................................1117
Resource Streamer Preemption Status ..................................................................................................1118
Restored Timestamp LSDW ..................................................................................................................1120
Restored timestamp MSDW ...................................................................................................................1121
RID_CC ................................................................................................................................................1122
RING_BUFFER_HEAD_PREEMPT_REG .................................................................................................1123
Ring Buffer Control ...............................................................................................................................1125
Ring Buffer Current Context ID Register ..............................................................................................1128
Ring Buffer Head ..................................................................................................................................1129
Ring Buffer Start ....................................................................................................................................1131
Ring Buffer Tail .......................................................................................................................................1132
Root Table Address Pointer Value First 31_0 .......................................................................................1134
Root Table Address Pointer Value Second 31_0 ...................................................................................1135
RP Decrease Limit ..................................................................................................................................1136
<table>
<thead>
<tr>
<th>Register</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>RP Down Timeout</td>
<td>1137</td>
</tr>
<tr>
<td>RP Downwards Evaluation Interval</td>
<td>1138</td>
</tr>
<tr>
<td>RP Increase Limit</td>
<td>1139</td>
</tr>
<tr>
<td>RP Normal Software Frequency Request</td>
<td>1140</td>
</tr>
<tr>
<td>RP Software Frequency Request Hysteresis</td>
<td>1141</td>
</tr>
<tr>
<td>RP Upwards Evaluation Interval</td>
<td>1142</td>
</tr>
<tr>
<td>RP Video Turbo Software Frequency Request</td>
<td>1143</td>
</tr>
<tr>
<td>RS_PREEMPT_STATUS_UDW</td>
<td>1144</td>
</tr>
<tr>
<td>RS Preemption Hint</td>
<td>1145</td>
</tr>
<tr>
<td>RS Preemption Hint UDW</td>
<td>1146</td>
</tr>
<tr>
<td>Sampler control register</td>
<td>1147</td>
</tr>
<tr>
<td>SAMPLER Mode Register</td>
<td>1148</td>
</tr>
<tr>
<td>SAMPLER READ DATA</td>
<td>1149</td>
</tr>
<tr>
<td>Save Timer</td>
<td>1150</td>
</tr>
<tr>
<td>SB_ADDRESS</td>
<td>1151</td>
</tr>
<tr>
<td>SB_DATA</td>
<td>1152</td>
</tr>
<tr>
<td>SB_REQ_TRIGGER</td>
<td>1153</td>
</tr>
<tr>
<td>SCPD0</td>
<td>1155</td>
</tr>
<tr>
<td>SCRATCH1</td>
<td>1156</td>
</tr>
<tr>
<td>SCRATCH for LNCFunit</td>
<td>1157</td>
</tr>
<tr>
<td>Scratch Register 1</td>
<td>1158</td>
</tr>
<tr>
<td>Scratch Register 2</td>
<td>1159</td>
</tr>
<tr>
<td>Second Buffer Size</td>
<td>1160</td>
</tr>
<tr>
<td>Second Level Batch Buffer Head Pointer Preemption Register</td>
<td>1161</td>
</tr>
<tr>
<td>Second Level Batch Buffer Head Pointer Register</td>
<td>1163</td>
</tr>
<tr>
<td>Second Level Batch Buffer State Register</td>
<td>1165</td>
</tr>
<tr>
<td>Second Level Batch Buffer Upper Head Pointer Preemption Register</td>
<td>1167</td>
</tr>
<tr>
<td>Second Level Batch Buffer Upper Head Pointer Register</td>
<td>1168</td>
</tr>
<tr>
<td>Semaphore Polling Interval on Wait</td>
<td>1169</td>
</tr>
<tr>
<td>SSID_SID</td>
<td>1170</td>
</tr>
<tr>
<td>Staggered EU/SAMPLER PAUSE on Frequency Change</td>
<td>1171</td>
</tr>
<tr>
<td>Storage 1</td>
<td>1172</td>
</tr>
<tr>
<td>Storage 2</td>
<td>1173</td>
</tr>
</tbody>
</table>
Command Reference: Registers

Storage 3 .................................................................................................................. 1174
Storage 4 .................................................................................................................. 1175
Storage 5 .................................................................................................................. 1176
Storage 6 .................................................................................................................. 1177
Storage 7 .................................................................................................................. 1178
Storage 8 .................................................................................................................. 1179
Storage 9 .................................................................................................................. 1180
Stream Output Num Primitives Written Counter ................................................. 1181
Stream Output Primitive Storage Needed Counters ............................................. 1182
Stream Output Write Offsets .................................................................................. 1183
SWF1 ....................................................................................................................... 1184
SWF2 ....................................................................................................................... 1185
SWF3 ....................................................................................................................... 1186
SWF4 ....................................................................................................................... 1187
SWF5 ....................................................................................................................... 1188
SWF6 ....................................................................................................................... 1189
SWF7 ....................................................................................................................... 1190
SWF8 ....................................................................................................................... 1191
SWF9 ....................................................................................................................... 1192
SWF10 ..................................................................................................................... 1193
SWF11 ..................................................................................................................... 1194
SWF12 ..................................................................................................................... 1195
SWF13 ..................................................................................................................... 1196
SWF14 ..................................................................................................................... 1197
SWF15 ..................................................................................................................... 1198
SWF16 ..................................................................................................................... 1199
SWF17 ..................................................................................................................... 1200
SWF18 ..................................................................................................................... 1201
SWF19 ..................................................................................................................... 1202
SWF20 ..................................................................................................................... 1203
SWF21 ..................................................................................................................... 1204
SWF22 ..................................................................................................................... 1205
SWF23 ..................................................................................................................... 1206
# Command Reference: Registers

<table>
<thead>
<tr>
<th>Register Name</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>SWF24</td>
<td>1207</td>
</tr>
<tr>
<td>SWF25</td>
<td>1208</td>
</tr>
<tr>
<td>SWF26</td>
<td>1209</td>
</tr>
<tr>
<td>SWF27</td>
<td>1210</td>
</tr>
<tr>
<td>SWF28</td>
<td>1211</td>
</tr>
<tr>
<td>SWF29</td>
<td>1212</td>
</tr>
<tr>
<td>SWF30</td>
<td>1213</td>
</tr>
<tr>
<td>SWF31</td>
<td>1214</td>
</tr>
<tr>
<td>SWF32</td>
<td>1215</td>
</tr>
<tr>
<td>SWF33</td>
<td>1216</td>
</tr>
<tr>
<td>SWF34</td>
<td>1217</td>
</tr>
<tr>
<td>SWF35</td>
<td>1218</td>
</tr>
<tr>
<td>SWF36</td>
<td>1219</td>
</tr>
<tr>
<td>SWSMISC1</td>
<td>1220</td>
</tr>
<tr>
<td>Thread Dispatched Count Register</td>
<td>1221</td>
</tr>
<tr>
<td>Thread Faulted Count Register</td>
<td>1222</td>
</tr>
<tr>
<td>Thread Fault Status Register 0</td>
<td>1223</td>
</tr>
<tr>
<td>Thread Fault Status Register 1</td>
<td>1224</td>
</tr>
<tr>
<td>Thread Load Status Register 0</td>
<td>1225</td>
</tr>
<tr>
<td>Thread Load Status Register 1</td>
<td>1226</td>
</tr>
<tr>
<td>Thread Mode Register</td>
<td>1227</td>
</tr>
<tr>
<td>Thread Restart Control Register</td>
<td>1230</td>
</tr>
<tr>
<td>TILECTL</td>
<td>1231</td>
</tr>
<tr>
<td>TiledResources Invalid Tile Detection Register</td>
<td>1232</td>
</tr>
<tr>
<td>Tiled Resources Translation Table Control Registers</td>
<td>1233</td>
</tr>
<tr>
<td>TiledResources VA Detection Registers</td>
<td>1234</td>
</tr>
<tr>
<td>Tiled Resources VA Translation Table L3 ptr - DW0</td>
<td>1235</td>
</tr>
<tr>
<td>Tiled Resources VA Translation Table L3 ptr - DW1</td>
<td>1236</td>
</tr>
<tr>
<td>TLB_RD_ADDRESS Register</td>
<td>1237</td>
</tr>
<tr>
<td>TLB_RD_DATA0 Register</td>
<td>1238</td>
</tr>
<tr>
<td>TLB_RD_DATA1 Register</td>
<td>1239</td>
</tr>
<tr>
<td>Transcode Attack Status Register</td>
<td>1240</td>
</tr>
<tr>
<td>TRNULLDETCT</td>
<td>1241</td>
</tr>
</tbody>
</table>
Valid Bit Vector 1 for L3 ................................................................. 1347
Valid Bit Vector 1 for MFX ............................................................ 1348
Valid Bit Vector 1 for MFX SL1 ...................................................... 1349
Valid Bit Vector 1 for MTTLB ......................................................... 1350
Valid Bit Vector 1 for MTVICTLB .................................................. 1351
Valid Bit Vector 1 for RCC ............................................................. 1352
Valid Bit Vector 1 for RCCTLB ...................................................... 1353
Valid Bit Vector 1 for RCZTLB ...................................................... 1354
Valid Bit Vector 1 for TLBPEND registers .................................... 1355
Valid Bit Vector 1 for VEBX ........................................................... 1356
Valid Bit Vector 1 for WIDI ........................................................... 1357
Valid Bit Vector 1 for Z ................................................................. 1358
Valid Bit Vector 2 for CVS ............................................................ 1359
Valid Bit Vector 2 for GAB ............................................................ 1360
Valid Bit Vector 2 for L3 ............................................................... 1361
Valid Bit Vector 2 for MFX ............................................................ 1362
Valid Bit Vector 2 for MFX SL1 ...................................................... 1363
Valid Bit Vector 2 for RCC ............................................................ 1364
Valid Bit Vector 2 for Z ................................................................. 1365
Valid Bit Vector 3 for CVS ............................................................ 1366
Valid Bit Vector 3 for L3 ............................................................... 1367
Valid Bit Vector 3 for MFX ............................................................ 1368
Valid Bit Vector 3 for MFX SL1 ...................................................... 1369
Valid Bit Vector 3 for RCC ............................................................ 1370
Valid Bit Vector 3 for Z ................................................................. 1371
Valid Bit Vector 4 for L3 ............................................................... 1372
Valid Bit Vector 4 for MFX ............................................................ 1373
Valid Bit Vector 4 for MFX SL1 ...................................................... 1374
Valid Bit Vector 4 for RCC ............................................................ 1375
Valid Bit Vector 4 for Z ................................................................. 1376
Valid Bit Vector 5 for L3 ............................................................... 1377
Valid Bit Vector 5 for MFX ............................................................ 1378
Valid Bit Vector 5 for MFX SL1 ...................................................... 1379
Valid Bit Vector 5 for RCC .................................................................................................................. 1380
Valid Bit Vector 5 for Z .......................................................................................................................... 1381
Valid Bit Vector 6 for L3 .......................................................................................................................... 1382
Valid Bit Vector 6 for MFX ...................................................................................................................... 1383
Valid Bit Vector 6 for MFX SL1 .............................................................................................................. 1384
Valid Bit Vector 6 for RCC ...................................................................................................................... 1385
Valid Bit Vector 6 for Z .......................................................................................................................... 1386
Valid Bit Vector 7 for L3 .......................................................................................................................... 1387
Valid Bit Vector 7 for MFX ...................................................................................................................... 1388
Valid Bit Vector 7 for MFX SL1 .............................................................................................................. 1389
Valid Bit Vector 7 for RCC ...................................................................................................................... 1390
Valid Bit Vector 7 for Z .......................................................................................................................... 1391
Valid Bit Vector 8 for L3 .......................................................................................................................... 1392
Valid Bit Vector 8 for Z .......................................................................................................................... 1393
Valid Bit Vector 9 for L3 .......................................................................................................................... 1394
Valid Bit Vector 9 for Z .......................................................................................................................... 1395
Valid Bit Vector 10 for L3 ....................................................................................................................... 1396
Valid Bit Vector 10 for Z ........................................................................................................................ 1397
Valid Bit Vector 11 for L3 ........................................................................................................................ 1398
Valid Bit Vector 11 for Z ........................................................................................................................ 1399
Valid Bit Vector 12 for L3 ....................................................................................................................... 1400
Valid Bit Vector 12 for Z ........................................................................................................................ 1401
Valid Bit Vector 13 for L3 ....................................................................................................................... 1402
Valid Bit Vector 13 for Z ........................................................................................................................ 1403
Valid Bit Vector 14 for L3 ....................................................................................................................... 1404
Valid Bit Vector 14 for Z ........................................................................................................................ 1405
Valid Bit Vector 15 for L3 ....................................................................................................................... 1406
Valid Bit Vector 15 for Z ........................................................................................................................ 1407
Valid Bit Vector 16 for L3 ....................................................................................................................... 1408
Valid Bit Vector 17 for L3 ....................................................................................................................... 1409
Valid Bit Vector 18 for L3 ....................................................................................................................... 1410
Valid Bit Vector 19 for L3 ....................................................................................................................... 1411
Valid Bit Vector 20 for L3 ....................................................................................................................... 1412
Command Reference: Registers

Valid Bit Vector 21 for L3 ................................................................. 1413
Valid Bit Vector 22 for L3 ................................................................. 1414
Valid Bit Vector 23 for L3 ................................................................. 1415
Valid Bit Vector for VLF ................................................................. 1416
Valid Bit Vector for VLFSI1 ............................................................. 1417
VC .............................................................................................. 1418
VCES Idle Switch Delay ................................................................. 1419
VCID ......................................................................................... 1420
VCS_PREEMPTION_HINT .............................................................. 1421
VCS_PREEMPTION_HINT_UDW ...................................................... 1423
VCS Context ID Preemption Hint .................................................. 1424
VCS Context Sizes ................................................................. 1425
VCS Context Timestamp Count .................................................... 1426
VCS Counter for the bit stream decode engine ................................ 1427
VCS Error Identity Register ............................................................ 1428
VCS Error Mask Register ............................................................... 1429
VCS Error Status Register ............................................................. 1430
VCS Execute Condition Code Register .......................................... 1431
VCS General Purpose Register ..................................................... 1432
VCS Hardware Status Mask Register ............................................. 1433
VCS IDLE Max Count .................................................................. 1434
VCS Idle Switch Delay ................................................................. 1435
VCS Instruction Parser Mode Register ......................................... 1436
VCS Interrupt Mask Register ........................................................ 1438
VCS Mode Register for Software Interface .................................... 1439
VCS Reported Timestamp Count .................................................. 1441
VCS Reset Control Register ......................................................... 1442
VCS Ring Buffer Next Context ID Register .................................... 1443
VCS Semaphore Polling Interval on Wait ........................................ 1444
VCS Threshold for the counter of bit stream decode engine .......... 1445
VCW Clock Count ........................................................................ 1446
VCW Internal Latency .................................................................. 1447
VCW Min Max Latency ................................................................. 1448
<table>
<thead>
<tr>
<th>Register Name</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>VCW Total Latency</td>
<td>1449</td>
</tr>
<tr>
<td>VCW XY position</td>
<td>1450</td>
</tr>
<tr>
<td>VEBOX TLB Control Register</td>
<td>1451</td>
</tr>
<tr>
<td>VEBX Context Element Descriptor (High Part)</td>
<td>1452</td>
</tr>
<tr>
<td>VEBX Context Element Descriptor (Low Part)</td>
<td>1453</td>
</tr>
<tr>
<td>VEBX Context Element Descriptor (Low Part)</td>
<td>1454</td>
</tr>
<tr>
<td>VEBX LRA 0</td>
<td>1455</td>
</tr>
<tr>
<td>VEBX LRA 1</td>
<td>1456</td>
</tr>
<tr>
<td>VEBX PDP0/PML4/PASID Descriptor (High Part)</td>
<td>1458</td>
</tr>
<tr>
<td>VEBX PDP0/PML4/PASID Descriptor (Low Part)</td>
<td>1459</td>
</tr>
<tr>
<td>VEBX PDP1 Descriptor Register (High Part)</td>
<td>1460</td>
</tr>
<tr>
<td>VEBX PDP1 Descriptor Register (Low Part)</td>
<td>1461</td>
</tr>
<tr>
<td>VEBX PDP2 Descriptor Register (High Part)</td>
<td>1462</td>
</tr>
<tr>
<td>VEBX PDP2 Descriptor Register (Low Part)</td>
<td>1463</td>
</tr>
<tr>
<td>VEBX PDP3 Descriptor Register (High Part)</td>
<td>1464</td>
</tr>
<tr>
<td>VEBX PDP3 Descriptor Register (Low Part)</td>
<td>1465</td>
</tr>
<tr>
<td>VECS_PREEMPTION_HINT</td>
<td>1466</td>
</tr>
<tr>
<td>VECS Context ID Preemption Hint</td>
<td>1468</td>
</tr>
<tr>
<td>VECS Context Timestamp Count</td>
<td>1469</td>
</tr>
<tr>
<td>VECS Counter for the Video Enhancement Engine</td>
<td>1470</td>
</tr>
<tr>
<td>VECS Error Identity Register</td>
<td>1471</td>
</tr>
<tr>
<td>VECS Error Mask Register</td>
<td>1472</td>
</tr>
<tr>
<td>VECS Error Status Register</td>
<td>1473</td>
</tr>
<tr>
<td>VECS General Purpose Register</td>
<td>1474</td>
</tr>
<tr>
<td>VECS Hardware Status Mask Register</td>
<td>1475</td>
</tr>
<tr>
<td>VECS IDLE Max Count</td>
<td>1476</td>
</tr>
<tr>
<td>VECS Instruction Parser Mode Register</td>
<td>1477</td>
</tr>
<tr>
<td>VECS Interrupt Mask Register</td>
<td>1478</td>
</tr>
<tr>
<td>VECS Mode Register for Software Interface</td>
<td>1479</td>
</tr>
<tr>
<td>VECS PREEMPTION HINT UDW</td>
<td>1481</td>
</tr>
<tr>
<td>VECS Reported Timestamp Count</td>
<td>1482</td>
</tr>
<tr>
<td>VECS Reset Control Register</td>
<td>1483</td>
</tr>
<tr>
<td>VECS Semaphore Polling Interval on Wait</td>
<td>1484</td>
</tr>
</tbody>
</table>
VECS Threshold for the Counter of Video Enhancement Engine ........................................1485
VEO Current Pipe 0 XY Register ..................................................................................1486
VEO DN Pipe 0 XY Register .........................................................................................1487
VEO DN Pipe 1 XY Register .........................................................................................1488
VEO DV Count Register ...............................................................................................1489
VEO DV Hold Register ................................................................................................1490
VEO Previous Pipe 0 XY Register ................................................................................1493
VF Scratch Pad ..............................................................................................................1494
VFW Credit Count Register ........................................................................................1496
VIC Virtual page Address Registers ...........................................................................1497
VIDEOBUSYCOUNTER ..............................................................................................1498
Video Enhancement Mode Register ..............................................................................1499
Video Mode Register ..................................................................................................1502
VS Invocation Counter .................................................................................................1505
Wait For Event and Display Flip Flags Register .........................................................1506
Wait For Event and Display Flip Flags Register 1 ......................................................1511
Walkers Fault Register ...............................................................................................1517
WD_WNIC_MSG_ADDR ..............................................................................................1518
WGBOX State Arbitration Priority Control .................................................................1519
WIDI LRA 0 ................................................................................................................1520
WIDI LRA 1 ................................................................................................................1521
WIDI TLB Control Register .........................................................................................1522
WRID_VALID_REG0 .....................................................................................................1523
WRID_VALID_REG1 .....................................................................................................1524
WRID_VALID_REG2 .....................................................................................................1525
Write Watermark .........................................................................................................1526
ZTLB LRA 0 ................................................................................................................1527
ZTLB LRA 1 ................................................................................................................1528
## AFLC - Advanced Features Length and Capabilities

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:26</td>
<td><strong>RESERVED</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td>25</td>
<td></td>
<td><strong>FLR_CAP</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 1b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Function Level Reset Capability (FLR_CAP):</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0: Function Level Reset is not supported</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1: Function Level Reset is supported</td>
</tr>
<tr>
<td>24</td>
<td></td>
<td><strong>TP_CAP</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 1b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Transactions Pending Capability (TP_CAP):</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0: Transactions Pending bit is not supported</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1: Transactions Pending bit is supported</td>
</tr>
<tr>
<td>23:16</td>
<td></td>
<td><strong>LENGTH</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 06h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Advanced Features Structure Length(LENGTH):</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The Advanced Features capability structure is</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6 bytes long.</td>
</tr>
<tr>
<td>15:8</td>
<td></td>
<td><strong>NXT_PTR</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Once</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Next Pointer (NXT_PTR):</td>
</tr>
</tbody>
</table>
**AFLC - Advanced Features Length and Capabilities**

Points to the next item in the list (B0=Vendor Capabilities ID). This register should be programmed by BIOS during boot-up. Once written, this register becomes Read_Only. This register can only be cleared by a Reset. Write once so capabilities list can be changed if needed.

<table>
<thead>
<tr>
<th>7:0</th>
<th>CAP_ID</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Default Value:</td>
</tr>
<tr>
<td></td>
<td>Access:</td>
</tr>
</tbody>
</table>

Capability Identifier (CAP_ID):
A value of 13h identifies that this PCI Function is capable of Advanced Features.
Advanced Scheduler Reset Request Messages

<table>
<thead>
<tr>
<th>ASSRREQ - Advanced Scheduler Reset Request Messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 0810Ch</td>
</tr>
</tbody>
</table>

Hardware (CS, VCS) initiated Advanced Scheduler reset request messages.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td><strong>Message Mask</strong>&lt;br&gt;Access: RO&lt;br&gt;Message Mask&lt;br&gt;In order to write to bits 15:0, the corresponding message mask bits must be written. For example, for bit 14 to be set, bit 30 needs to be 1 : 40004000</td>
</tr>
<tr>
<td>15:4</td>
<td>Reserved</td>
<td>Access: RO&lt;br&gt;Reserved</td>
</tr>
<tr>
<td>3</td>
<td>VINunit cmfxrst reset request message (2nd Vbox)&lt;br&gt;Access: R/W Set&lt;br&gt;CMFX Reset Request Message from the VINunit in 2nd Vbox:&lt;br&gt;'1' : CMFX Reset Requested&lt;br&gt;- This bit is cleared by the CP upon completion of the reset request&lt;br&gt;'0' : CMFX Reset Not Requested</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>VINunit cmfxrst Reset Request message&lt;br&gt;Access: R/W Set&lt;br&gt;CMFX Reset Request Message from the VINunit:&lt;br&gt;'1' : CMFX Reset Requested&lt;br&gt;- This bit is cleared by the CP upon completion of the reset request&lt;br&gt;'0' : CMFX Reset Not Requested</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Render AS Reset Request Message&lt;br&gt;Access: R/W Set&lt;br&gt;Render AS Reset Request Message from the CSUnit:&lt;br&gt;'1' : Render AS Reset Requested&lt;br&gt;- This bit is cleared by the CP upon completion of the reset request&lt;br&gt;'0' : Render AS Reset Not Requested</td>
<td></td>
</tr>
</tbody>
</table>
# ASSRREQ - Advanced Scheduler Reset Request Messages

<table>
<thead>
<tr>
<th>Address</th>
<th>Media AS Reset Request Message</th>
<th>Access:</th>
<th>R/W Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Media AS Reset Request Message</td>
<td>R/W Set</td>
<td></td>
</tr>
</tbody>
</table>

- Media AS Reset Request Message from the VCSunit:
  - '1': Media AS Reset Requested
  - '0': Media AS Reset Not Requested

- This bit is cleared by the CP upon completion of the reset request
## Aggregate_Perf_Counter_A31

<table>
<thead>
<tr>
<th><strong>OAPERF_A31 - Aggregate_Perf_Counter_A31</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> BSpec</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 028F8h</td>
</tr>
<tr>
<td><strong>Valid Projects:</strong> [CHV, BSW]</td>
</tr>
</tbody>
</table>

This register reflects the count value of the OA Performance counter A31.

<table>
<thead>
<tr>
<th><strong>DWord</strong></th>
<th><strong>Bit</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Considerations</strong></td>
</tr>
</tbody>
</table>

This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.
# Aggregate_Perf_Counter_A32

<table>
<thead>
<tr>
<th><strong>OAPERF_A32 - Aggregate_Perf_Counter_A32</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space</strong>: MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project</strong>: CHV, BSW</td>
</tr>
<tr>
<td><strong>Source</strong>: BSpec</td>
</tr>
<tr>
<td><strong>Default Value</strong>: 0x00000000</td>
</tr>
<tr>
<td><strong>Access</strong>: R/W</td>
</tr>
<tr>
<td><strong>Size (in bits)</strong>: 32</td>
</tr>
<tr>
<td><strong>Address</strong>: 02900h</td>
</tr>
<tr>
<td><strong>Valid Projects</strong>: [CHV, BSW]</td>
</tr>
</tbody>
</table>

This register reflects the count value of the OA Performance counter A32

<table>
<thead>
<tr>
<th><strong>DWord</strong></th>
<th><strong>Bit</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Considerations</strong>&lt;br&gt;&lt;br&gt;<strong>Format</strong>: U32&lt;br&gt;&lt;br&gt;This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no &quot;latch and hold&quot; mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.</td>
</tr>
</tbody>
</table>
**Aggregate_Perf_Counter_A33**

<table>
<thead>
<tr>
<th><strong>OAPERF_A33 - Aggregate_Perf_Counter_A33</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: BSpec</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Access: R/W</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 02904h</td>
</tr>
<tr>
<td>Valid Projects: [CHV, BSW]</td>
</tr>
</tbody>
</table>

This register reflects the count value of the OA Performance counter A33

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Considerations</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format:</strong> U32</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no &quot;latch and hold&quot; mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.</td>
</tr>
</tbody>
</table>
## Aggregate_Perf_Counter_A34

<table>
<thead>
<tr>
<th>OAPERF_A34 - Aggregate_Perf_Counter_A34</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> BSpec</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 02908h</td>
</tr>
<tr>
<td><strong>Valid Projects:</strong> [CHV, BSW]</td>
</tr>
</tbody>
</table>

This register reflects the count value of the OA Performance counter A34

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Considerations</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format:</strong> U32</td>
</tr>
</tbody>
</table>

This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.
## Aggregate_Perf_Counter_A35

<table>
<thead>
<tr>
<th>OAPERF_A35 - Aggregate_Perf_Counter_A35</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> BSpec</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 0290Ch</td>
</tr>
<tr>
<td><strong>Valid Projects:</strong> [CHV, BSW]</td>
</tr>
</tbody>
</table>

This register reflects the count value of the OA Performance counter A35

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Considerations</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format:</strong> U32</td>
</tr>
</tbody>
</table>
|       |       | This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.

---
# Aggregate Perf Counter A1

<table>
<thead>
<tr>
<th><strong>OAPERF_A1 - Aggregate Perf Counter A1</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> BSpec</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 02808h</td>
</tr>
<tr>
<td><strong>Valid Projects:</strong> [CHV, BSW]</td>
</tr>
</tbody>
</table>

This register reflects the count value of the OA Performance counter A1. DefaultValue="00000000h"

<table>
<thead>
<tr>
<th><strong>DWord</strong></th>
<th><strong>Bit</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>Considerations: This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no &quot;latch and hold&quot; mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.</td>
</tr>
</tbody>
</table>
## Aggregate Perf Counter A5

### OAPERF_A5 - Aggregate Perf Counter A5

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>BSpec</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>02828h</td>
</tr>
<tr>
<td>Valid Projects:</td>
<td>[CHV, BSW]</td>
</tr>
</tbody>
</table>

This register reflects the count value of the OA Performance counter A5. DefaultValue="00000000h"

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
</table>
| 0     | 31:0| **Considerations**  
This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.
### Aggregate Perf Counter A7

<table>
<thead>
<tr>
<th><strong>OAPERF_A7 - Aggregate Perf Counter A7</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> BSpec</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 02838h</td>
</tr>
<tr>
<td><strong>Valid Projects:</strong> [CHV, BSW]</td>
</tr>
</tbody>
</table>

This register reflects the count value of the OA Performance counter A7. DefaultValue="00000000h"

<table>
<thead>
<tr>
<th><strong>DWord</strong></th>
<th><strong>Bit</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Considerations</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no &quot;latch and hold&quot; mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.</td>
</tr>
</tbody>
</table>
### Aggregate Perf Counter A8

<table>
<thead>
<tr>
<th><strong>OAPERF_A8 - Aggregate Perf Counter A8</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> BSpec</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
</tbody>
</table>

**Address:** 02840h

**Valid Projects:** [CHV, BSW]

This register reflects the count value of the OA Performance counter A8. Default Value="00000000h"

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
</table>
| 0     | 31:0| **Considerations**  
This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads. |
### Aggregate Perf Counter A9

<table>
<thead>
<tr>
<th>OAPERF_A9 - Aggregate Perf Counter A9</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> BSpec</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 02848h</td>
</tr>
<tr>
<td><strong>Valid Projects:</strong> [CHV, BSW]</td>
</tr>
</tbody>
</table>

This register reflects the count value of the OA Performance counter A9. DefaultValue="00000000h"

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
</table>
| 0     | 31:0| **Considerations**  
This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads. |
# Aggregate Perf Counter A10

<table>
<thead>
<tr>
<th>OAPERF_A10 - Aggregate Perf Counter A10</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> BSpec</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 02850h</td>
</tr>
<tr>
<td><strong>Valid Projects:</strong> [CHV, BSW]</td>
</tr>
</tbody>
</table>

This register reflects the count value of the OA Performance counter A10. DefaultValue="00000000h"

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
</table>
| 0     | 31:0| **Considerations**  
  This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads. |
# Aggregate Perf Counter A11

## OAPERF_A11 - Aggregate Perf Counter A11

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>BSpec</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x0000000000h</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>02858h</td>
</tr>
<tr>
<td>Valid Projects:</td>
<td>[CHV, BSW]</td>
</tr>
</tbody>
</table>

This register reflects the count value of the OA Performance counter A11. Default Value = "00000000h"

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
</table>
| 0     | 31:0| **Considerations**  
This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.
## Aggregate Perf Counter A12

<table>
<thead>
<tr>
<th>OAPERF_A12 - Aggregate Perf Counter A12</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> BSpec</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 02860h</td>
</tr>
<tr>
<td><strong>Valid Projects:</strong> [CHV, BSW]</td>
</tr>
</tbody>
</table>

This register reflects the count value of the OA Performance counter A12. DefaultValue="00000000h"

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
</table>
| 0     | 31:0| **Considerations**  
This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.
Aggregates Perf Counter A13

<table>
<thead>
<tr>
<th>OAPERF_A13 - Aggregate Perf Counter A13</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> BSpec</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 02868h</td>
</tr>
<tr>
<td><strong>Valid Projects:</strong> [CHV, BSW]</td>
</tr>
</tbody>
</table>

This register reflects the count value of the OA Performance counter A13. DefaultValue=“00000000h”

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
</table>
| 0     | 31:0| **Considerations**
|       |     | This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.
### Aggregate Perf Counter A14

<table>
<thead>
<tr>
<th>OAPERF_A14 - Aggregate Perf Counter A14</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space:</td>
</tr>
<tr>
<td>Project:</td>
</tr>
<tr>
<td>Source:</td>
</tr>
<tr>
<td>Default Value:</td>
</tr>
<tr>
<td>Access:</td>
</tr>
<tr>
<td>Size (in bits):</td>
</tr>
<tr>
<td>Address:</td>
</tr>
<tr>
<td>Valid Projects:</td>
</tr>
</tbody>
</table>

This register reflects the count value of the OA Performance counter A14. DefaultValue="00000000h"

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Considerations</strong>&lt;br&gt; This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no &quot;latch and hold&quot; mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.</td>
</tr>
</tbody>
</table>
Aggregate Perf Counter A15

<table>
<thead>
<tr>
<th>OAPERF_A15 - Aggregate Perf Counter A15</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> BSpec</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 02878h</td>
</tr>
<tr>
<td><strong>Valid Projects:</strong> [CHV, BSW]</td>
</tr>
</tbody>
</table>

This register reflects the count value of the OA Performance counter A15. Default Value="00000000h"

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
</table>
| 0     | 31:0 | **Considerations**
This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.
### Aggregate Perf Counter A16

**OAPERF_A16 - Aggregate Perf Counter A16**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>BSpec</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>02880h</td>
</tr>
<tr>
<td>Valid Projects:</td>
<td>[CHV, BSW]</td>
</tr>
</tbody>
</table>

This register reflects the count value of the OA Performance counter A16. DefaultValue="00000000h"

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Considerations</strong>&lt;br&gt;This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no &quot;latch and hold&quot; mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.</td>
</tr>
</tbody>
</table>
### Aggregate Perf Counter A17

<table>
<thead>
<tr>
<th><strong>OAPERF_A17 - Aggregate Perf Counter A17</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> BSpec</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 02888h</td>
</tr>
<tr>
<td><strong>Valid Projects:</strong> [CHV, BSW]</td>
</tr>
</tbody>
</table>

This register reflects the count value of the OA Performance counter A17. DefaultValue=“00000000h”

<table>
<thead>
<tr>
<th><strong>DWord</strong></th>
<th><strong>Bit</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Considerations</strong></td>
</tr>
</tbody>
</table>

This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.
Aggregate Perf Counter A18

<table>
<thead>
<tr>
<th>OAPERF_A18 - Aggregate Perf Counter A18</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: BSpec</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Access: R/W</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 02890h</td>
</tr>
<tr>
<td>Valid Projects: [CHV, BSW]</td>
</tr>
</tbody>
</table>

This register reflects the count value of the OA Performance counter A9. DefaultValue="00000000h"

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
</table>
| 0     | 31:0| **Considerations**  
This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads. |
## Aggregate Perf Counter A21

<table>
<thead>
<tr>
<th><strong>OAPERF_A21 - Aggregate Perf Counter A21</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong></td>
</tr>
<tr>
<td><strong>Project:</strong></td>
</tr>
<tr>
<td><strong>Source:</strong></td>
</tr>
<tr>
<td><strong>Default Value:</strong></td>
</tr>
<tr>
<td><strong>Access:</strong></td>
</tr>
<tr>
<td><strong>Size (in bits):</strong></td>
</tr>
<tr>
<td><strong>Address:</strong></td>
</tr>
<tr>
<td><strong>Valid Projects:</strong></td>
</tr>
</tbody>
</table>

This register reflects the count value of the OA Performance counter A21. DefaultValue="00000000h"

<table>
<thead>
<tr>
<th><strong>DWord</strong></th>
<th><strong>Bit</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
</table>
| 0         | 31:0    | **Considerations**  
This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads. |
## Aggregate Perf Counter A22

<table>
<thead>
<tr>
<th>OAPERF_A22 - Aggregate Perf Counter A22</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> BSpec</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 028B0h</td>
</tr>
<tr>
<td><strong>Valid Projects:</strong> [CHV, BSW]</td>
</tr>
</tbody>
</table>

This register reflects the count value of the OA Performance counter A22. DefaultValue="00000000h"

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
</table>
| 0     | 31:0| **Considerations**
|       |     | This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads. |
## Aggregate Perf Counter A23

<table>
<thead>
<tr>
<th>OAPERF_A23 - Aggregate Perf Counter A23</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> BSpec</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 028B8h</td>
</tr>
<tr>
<td><strong>Valid Projects:</strong> [CHV, BSW]</td>
</tr>
</tbody>
</table>

This register reflects the count value of the OA Performance counter A23. Default Value = "00000000h"

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Considerations</strong>&lt;br&gt;This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no &quot;latch and hold&quot; mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.</td>
</tr>
</tbody>
</table>
## Aggregate Perf Counter A24

<table>
<thead>
<tr>
<th>OAPERF_A24 - Aggregate Perf Counter A24</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> BSpec</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 028C0h</td>
</tr>
<tr>
<td><strong>Valid Projects:</strong> [CHV, BSW]</td>
</tr>
</tbody>
</table>

This register reflects the count value of the OA Performance counter A24. DefaultValue="00000000h"

### DWord Bit Description

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Considerations</strong>&lt;br&gt;This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no &quot;latch and hold&quot; mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.</td>
</tr>
</tbody>
</table>
## Aggregate Perf Counter A25

<table>
<thead>
<tr>
<th><strong>OAPERF_A25 - Aggregate Perf Counter A25</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: BSpec</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Access: R/W</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 028C8h</td>
</tr>
<tr>
<td>Valid Projects: [CHV, BSW]</td>
</tr>
</tbody>
</table>

This register reflects the count value of the OA Performance counter A25. DefaultValue="00000000h"

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
</table>
| 0     | 31:0| **Considerations**  
This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads. |
Aggregate Perf Counter A26

### OAPERF_A26 - Aggregate Perf Counter A26

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>BSpec</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>028D0h</td>
</tr>
<tr>
<td>Valid Projects:</td>
<td>[CHV, BSW]</td>
</tr>
</tbody>
</table>

This register reflects the count value of the OA Performance counter A26. DefaultValue="00000000h"

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Considerations</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no &quot;latch and hold&quot; mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.</td>
</tr>
</tbody>
</table>
## Aggregate Perf Counter A27

<table>
<thead>
<tr>
<th><strong>OAPERF_A27 - Aggregate Perf Counter A27</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: BSpec</td>
</tr>
<tr>
<td>Default Value: 0x000000000</td>
</tr>
<tr>
<td>Access: R/W</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 028D8h</td>
</tr>
<tr>
<td>Valid Projects: [CHV, BSW]</td>
</tr>
</tbody>
</table>

This register reflects the count value of the OA Performance counter A27. DEFAULTVALUE="00000000h"

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
</table>
| 0     | 31:0| **Considerations**  
This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.
### Aggregate Perf Counter A28

<table>
<thead>
<tr>
<th>OAPERF_A28 - Aggregate Perf Counter A28</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space:</td>
</tr>
<tr>
<td>MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project:</td>
</tr>
<tr>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
</tr>
<tr>
<td>BSpec</td>
</tr>
<tr>
<td>Default Value:</td>
</tr>
<tr>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
</tr>
<tr>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
</tr>
<tr>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
</tr>
<tr>
<td>028E0h</td>
</tr>
<tr>
<td>Valid Projects:</td>
</tr>
<tr>
<td>[CHV, BSW]</td>
</tr>
</tbody>
</table>

This register reflects the count value of the OA Performance counter A28. DefaultValue="00000000h"

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Considerations</strong>&lt;br&gt;This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no &quot;latch and hold&quot; mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.</td>
</tr>
</tbody>
</table>
## Aggregate Perf Counter A29

<table>
<thead>
<tr>
<th>OAPERF_A29 - Aggregate Perf Counter A29</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: BSpec</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Access: R/W</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 028E8h</td>
</tr>
<tr>
<td>Valid Projects: [CHV, BSW]</td>
</tr>
</tbody>
</table>

This register reflects the count value of the OA Performance counter A29. DefaultValue="00000000h"

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Considerations</strong>&lt;br&gt;This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no &quot;latch and hold&quot; mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.</td>
</tr>
</tbody>
</table>
## Aggregate Perf Counter A30

<table>
<thead>
<tr>
<th>OAPERF_A30 - Aggregate Perf Counter A30</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> BSpec</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 028F0h</td>
</tr>
<tr>
<td><strong>Valid Projects:</strong> [CHV, BSW]</td>
</tr>
</tbody>
</table>

This register reflects the count value of the OA Performance counter A30. Default Value = "00000000h"

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
</table>
| 0     | 31:0| **Considerations**  
This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads. |
### AFCTLSTS

**AFCTLSTS - AFCTLSTS**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>PCI: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>16</td>
</tr>
<tr>
<td>Address:</td>
<td>000A8h</td>
</tr>
</tbody>
</table>

**FLR control Advanced Feature Status**

#### DWord Bit Description

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>15:9</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved (RSVD)</td>
</tr>
</tbody>
</table>

| 8     | TP  | Default Value: 0b |
|       |     | Access: RO |

**Transaction Pending (TP):**

1: The Function has issued one or more non-posted transactions which have not been completed, including non-posted transactions that a target has terminated with Retry.

0: All non-posted transactions have been completed.

| 7:1   | Reserved |
|       | Default Value: 00h |
|       | Access: RO |
|       | Reserved (RSVD) |

<table>
<thead>
<tr>
<th>0</th>
<th>INIT_FLR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td>Access: R/W Set</td>
</tr>
</tbody>
</table>

**Initiate Function Level Reset (INIT_FLR):**

A write of 1b initiates Function Level Reset (FLR).

FLR requirements are defined in the PCI Express Base Specification. Registers and state information that do not apply to conventional PCI are exempt from the FLR requirements given there.

Once written 1, FLR will be initiated. During FLR, a read will return 1’s since device 2 reads abort. Once FLR completes, hardware will clear the bit to 0.
## All Engine Fault Register

### Register Information

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space</td>
<td>MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits)</td>
<td>32</td>
</tr>
<tr>
<td>Address</td>
<td>04094h</td>
</tr>
</tbody>
</table>

### Bit Descriptions

**All Engine Fault Register**

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:1</td>
<td><strong>All Engine Fault Register</strong></td>
</tr>
</tbody>
</table>

- **Default Value:** 0000000000000000000000000000000b
- **Access:** R/W

- Bit[31:15]: Reserved.
- Bit[14:12]: Engine ID:
  - 000b - GFX.
  - 001b - MFX0.
  - 010b - MFX1.
  - 011b - VEBX.
  - 100b - BLT.
  - 110b - WIDI.
- Bit[11]: Reserved.
- Bit[10:3]: SRCID of Fault.
  - This is the Source ID of the unit that requested the cycle that generated the First Page fault for this engine. This value is locked and not updated on subsequent faults, until the valid bit of this register is cleared by SW.
- Bit[2:1]: Fault Type (GFX_FT):
  - 00b - Invalid PTE Fault.
  - 01b - Invalid PDE Fault.
  - 10b - Invalid PDPE Fault.
  - 11b - Invalid PML4E Fault.
  - This value is locked and not updated on subsequent faults, until the valid bit of this register is cleared by SW.
- All bits are only valid with bit[0]=1.

**Valid Bit**

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td><strong>Valid Bit</strong></td>
</tr>
</tbody>
</table>

- **Default Value:** 0b
- **Access:** R/W

- This bit indicates that the first fault for this engine has been recorded. It can only be cleared by SW, which also clears the other fields.
# ARAT CUTRIG HI

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0A1B4h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>ARAT TRIG HI</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[31:0]: 63:32 of (ARAT_TDELTA + TSC)[63:32] when A174[0] transitions to a one, or when A174[1:0]=11 and TSC crosses over ARAT_TRIG</td>
</tr>
</tbody>
</table>
## ARAT CUTRIG LO

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>31:3</td>
<td>ARAT TRIG LO</td>
<td>RO</td>
</tr>
<tr>
<td>2</td>
<td>Reserved</td>
<td>RO</td>
</tr>
<tr>
<td>1:0</td>
<td>ARAT TRIG Mode</td>
<td>RO</td>
</tr>
</tbody>
</table>

**Register Space:** MMIO: 0/2/0  
**Project:** CHV, BSW  
**Source:** PRM  
**Default Value:** 0x00000000  
**Size (in bits):** 32  
**Address:** 0A1B0h

### ARAT CUTRIG LO - ARAT CUTRIG LO

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:3</td>
<td>ARAT TRIG LO</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>31:3 of (ARAT_TDELTA + TSC)[31:3] when A174[0] transitions to a one, or when A174[1:0]=11 and TSC crosses over ARAT_TRIG</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>Reserved</td>
<td>RO</td>
</tr>
<tr>
<td>1:0</td>
<td></td>
<td>ARAT TRIG Mode</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0xA174[1:0] when A174[0] transitions to a one, or when A174[1:0]=11 and TSC crosses over ARAT_TRIG</td>
<td></td>
</tr>
</tbody>
</table>
## ARAT Delta (LSB)

### ARAT_TDELTAL_LOW - ARAT Delta (LSB)

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0A174h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:3</td>
<td><strong>Lower Bits of Delta Time for ARAT</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Low Bits [31:3] of Delta Time, in 80ns increments (LSB would be in 10ns increments, if it went down to zero)</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td><strong>ARAT Mode</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0b : One-Shot Mode (default)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1b : Periodic Mode</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td><strong>ARAT Enable</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0b : ARAT Disabled (default)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1b : ARAT Enabled</td>
</tr>
</tbody>
</table>
## ARAT Delta (MSB)

### ARAT_TDELTAL_HIGH - ARAT Delta (MSB)

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0A170h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>23:0</td>
<td></td>
<td><strong>Upper Bits of Delta Time for ARAT</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>High Bits [55:32] of Delta Time, in 80ns increments</td>
</tr>
</tbody>
</table>
### ARAT_POSTCUBUSY - ARAT POST CU BUSY

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:3</td>
<td>ARAT PostThreshold</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PostThreshold, in 80ns</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>increments to prevent short</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>gfx_clockstartreq. If</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>current_TSC - ARAT_CUTRIG &gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>PostThreshold, show GFX as</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>busy.</td>
<td></td>
</tr>
<tr>
<td>2:0</td>
<td></td>
<td>Reserved</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access:</td>
<td></td>
</tr>
</tbody>
</table>

Register Space: MMIO: 0/2/0  
Project: CHV, BSW  
Source: PRM  
Default Value: 0x00000000  
Size (in bits): 32  
Address: 0A17Ch
## ARAT PRE CU BUSY

### ARAT_PRECUBUSY - ARAT PRE CU BUSY

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0A178h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:3</td>
<td><strong>ARAT PreThreshold</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PreThreshold, in 80ns increments to prevent short gfx_clockstartreq. If ARAT_CUTRIG minus current_TSC &lt; PreThreshold, show GFX as busy.</td>
</tr>
<tr>
<td>2:0</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
</tbody>
</table>

*ARAT PRE CU BUSY*
# Arbiter Control Register

## GARBCNTLREG - Arbiter Control Register

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x29124100 CHV, BSW</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0B004h</td>
</tr>
</tbody>
</table>

### DWord 0

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>31</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td>30</td>
<td><strong>Disables hashing function</strong></td>
</tr>
</tbody>
</table>

- **Access:** R/W
- Disables hashing function to generate bank_id[1:0] for L3$ bank accessing, and forces the use of address[7:6] for bank_id[1:0]. (default) Hash function enabled to generate L3$ bank IDs.
- 1: L3$ address[7:6] used as L3$ bank IDs.
- **lncf_csr_l3bankidhashdis.**
  (This bit needs to set corresponding bit lpfcon_csr_l3bankidhashdis in LPFC.)

### DWord 29:28

#### Arbitration priority order between RCC and MSC

- **Default Value:** 10b
- **Access:** R/W

Arbitration priority order between RCC and MSC.
- 00b/11b: Invalid; default setting used.
- 10b: Default setting; RCC MSC (i.e., MSC has higher priority).
- 01b: RCC MSC (i.e., RCC has higher priority).
- **lncf_csr_rcc_msc_pri[1:0].**

### DWord 27:22

#### Arbitration priority order between RCZ, STC, and HIZ

- **Default Value:** 100100b
- **Access:** R/W

Arbitration priority order between RCZ, STC, and HIZ.
- 100100b: Default setting; RCZ STC HIZ.
- (i.e., RCZ has lowest priority; HIZ has highest priority).
- 100011b: RCZ ; HIZ ; STC.
- 011000b: STC ; RCZ ; HIZ.
- 010010b: STC ; HIZ ; RCZ.
- 001001b: HIZ ; RCZ ; STC.
- 000110b: HIZ ; STC ; RCZ.
- **Note:** Others settings are invalid, and result in use of default.
- **lncf_csr_rcz_stc_hiz_pri[5:0].**
### GARBCNTLREG - Arbiter Control Register

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>21:19</td>
<td><strong>Write data port arbitration priority between Z client writes and L3$ evictions</strong></td>
<td>010b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>Z Max Write Request Limit Count (GFXC_MRLC). This is the MAX number of Allowed Requests Count - These counters keep track of the accepted requests from each engine. Requests are counted, regardless of kind of cycle (both Slice 0 and 1). Minimum count value must be = 1. lncf_csr_wdpagapz[2:0].</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18:16</td>
<td><strong>Write data port arbitration priority between C client writes and Z/L3$ writes/evictions</strong></td>
<td>010b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>C Max Request Limit Count (GFXZ_MRLC). This is the MAX number of Allowed Requests Count - These counters keep track of the accepted requests from each engine. Requests are counted, regardless of kind of cycle (both Slice 0 and 1). Minimum count value must be = 1. lncf_csr_wdpagapc[2:0].</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td><strong>Reserved</strong></td>
<td></td>
<td>RO</td>
</tr>
<tr>
<td>14:12</td>
<td><strong>L3 Max Write Request Limit Count</strong></td>
<td>100b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>L3 Max Write Request Limit Count (GFXL3_MRLC). This is the MAX number of Allowed Requests Count - These counters keep track of the accepted requests from each engine. Requests are counted, regardless of kind of cycle (Miss/Hit/ Present). Minimum count value must be = 1. lncf_csr_wdpagapl3[2:0].</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11:9</td>
<td><strong>Reserved</strong></td>
<td></td>
<td>RO</td>
</tr>
<tr>
<td>8</td>
<td><strong>GAPs_fixarb_en</strong></td>
<td>1b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>GAPs_fixarb_en</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6:0</td>
<td><strong>Reserved</strong></td>
<td></td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td>Project: CHV, BSW</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Arbiter Mode Control Register

<table>
<thead>
<tr>
<th>Register Space</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits)</td>
<td>32</td>
</tr>
<tr>
<td>Address</td>
<td>04030h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td><strong>Mask Bits</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0000000000000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Mask Bits act as Write Enables for the bits[15:0] of this register.</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td><strong>Extra Register Bit 15</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit 15 toggles (XOR) the meaning of Per Client Write Drop Enables (Register 40b4); If 0, drop per client happens as stated in register 40b4 definition; If 1, the meaning changes, and a 1 on a bit in register 40b4 means dont drop while 0 means drop. In this case, the default (for clients not included in 40b4) will be drop enabled.</td>
</tr>
<tr>
<td>14</td>
<td></td>
<td><strong>Extra Register Bit 14</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PD load disable - When this bit is set, the PD load is disabled for GFX/MFX0/MFX1.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-step: Default Value: 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B-step: Default Value: 0 - Bug ID: 1905990</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Future steppings can have value 1.</td>
</tr>
<tr>
<td>13</td>
<td></td>
<td><strong>DC GDR</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td><strong>HIZ GDR</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td><strong>STC GDR</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td>Register</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td><strong>BLB GDR</strong></td>
<td>10-bit register for controlling the Arbiter Mode. Default value: 0b. Access: R/W.</td>
<td></td>
</tr>
<tr>
<td><strong>GAM PD GDR</strong></td>
<td>9-bit register for controlling the Arbiter Mode. Default value: 0b. Access: R/W.</td>
<td></td>
</tr>
<tr>
<td><strong>Extra Register Bit 8</strong></td>
<td>8-bit register for controlling the Arbiter Mode. Default value: 0b. Access: R/W.</td>
<td></td>
</tr>
<tr>
<td><strong>Cacheability Attribute Override</strong></td>
<td>Override for cacheability attributes. Default value: 00b. Access: R/W.</td>
<td></td>
</tr>
<tr>
<td><strong>Extra Register Bit 5</strong></td>
<td>5-bit register for controlling the Arbiter Mode. Default value: 0b. Access: R/W.</td>
<td></td>
</tr>
<tr>
<td><strong>VMC GDR Enable</strong></td>
<td>4-bit register for enabling the VMC GDR algorithm. Default value: 0b. Access: R/W.</td>
<td></td>
</tr>
<tr>
<td><strong>Texture Cache (MT) GDR Enable Bit</strong></td>
<td>3-bit register for enabling the Texture Cache (MT) GDR algorithm. Default value: 0b. Access: R/W.</td>
<td></td>
</tr>
</tbody>
</table>

### Description

Snoop Override [CHV, BSW Only]
0 - No override (default)
1 - Snoop is set for all accesses to memory

Cacheability Attribute Override
00b No override.
01b UC (LLC/eLLC) - Allocation age is don't care.
10b WT in LLC/eLLC - Aged is 3.
11b WB in LLC/eLLC - Aged is 3.
The above conditions apply for the following conditions only:
1. Register overwrite except for GTT, CFG and L3 coherent wcil cycles
2. Read- GTTRD, CFGRD
3. Write- GTTWR, CFGWR, DMWR (with gam_ci_wcoherenttype[2:0]="001" WCIL* w/self snoop)

Extra Register Bit 5
Reserved.

When this bit is set, data requested from the VMC client is generated by the GDR Algorithm.

When this bit is set, data requested from the Texture Cache (MT) client is generated by the GDR algorithm.
## ARB_MODE - Arbiter Mode Control Register

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td><strong>Depth (RCZ) Cache GDR Enable bit</strong></td>
<td>0b</td>
<td>R/W</td>
<td>Depth Cache GDR enable bit. Project: All. Format: U1. When this bit is set, data requested from the Depth Cache client is generated by the GDR algorithm.</td>
</tr>
<tr>
<td>1</td>
<td><strong>Color Cache (RCC) GDR Enable Bit</strong></td>
<td>0b</td>
<td>R/W</td>
<td>When this bit is set, data requested from the Color Cache (RCC) client is generated by the GDR algorithm.</td>
</tr>
<tr>
<td>0</td>
<td><strong>GTT Accesses GDR</strong></td>
<td>0b</td>
<td>R/W</td>
<td>When this bit is enabled along with the Client’s GDR bit, PPGTT and GGTT requests for this memory access are also tagged as GDR to SQ.</td>
</tr>
</tbody>
</table>
## ASLS

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>SCRATCH</td>
</tr>
</tbody>
</table>

This software scratch register only needs to be read/write accessible. The exact bit register usage must be worked out in common between System BIOS and driver software, but storage for switching/indicating up to 6 devices is possible with this amount.

For each device, the ASL control method with require two bits for _DOD (BIOS detectable yes or no, VGA/NonVGA), one bit for _DGS (enable/disable requested), and two bits for _DCS (enabled now/disabled now, connected or not).
### ASYNC_SLICE_COUNT - ASYNC_SLICE_COUNT

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000280</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0A204h</td>
</tr>
</tbody>
</table>

Async Slice Count Select Register

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:11</td>
<td><strong>Reserved</strong></td>
<td>00000h</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10:8</td>
<td></td>
<td><strong>Async_SS</strong></td>
<td>010b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Number of subslices to power (Async Mode):</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>001 : 1 subslice</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>010 : 2 subslices (GT1-based CHV, BSW only)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Actual SSS used = Async_SS if SScountEn=0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Actual SSS used = SScount if SScountEn=1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7:4</td>
<td></td>
<td><strong>Async_EU</strong></td>
<td>1000b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Maximum number of EUs to power per subslice if multiple subslices enabled Async Mode</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Actual EUs used = Async_EU if EUmin &lt; Async_EU &lt; EUmax</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Actual EUs used = EUmax if Async_EU &gt;= EUmax</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Actual EUs used = EUmin if EUmin &gt;= Async_EU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td><strong>Reserved</strong></td>
<td>0b</td>
<td>RO</td>
</tr>
<tr>
<td>2:0</td>
<td></td>
<td><strong>temp_slicecount</strong></td>
<td>000b</td>
<td>R/W</td>
</tr>
</tbody>
</table>

Slice Count Request in Asynchronous Mode.
# Auto Draw End Offset

<table>
<thead>
<tr>
<th><strong>3DPRIM_END_OFFSET - Auto Draw End Offset</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> RenderCS</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 02420h-02423h</td>
</tr>
<tr>
<td><strong>Valid Projects:</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>End Offset</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: U32</td>
</tr>
</tbody>
</table>

This register is used to store the end offset value used by the Vertex Fetch to determine when to stop processing the 3D_PRIMITIVE command. This register is valid when the End Offset Enable is set in the 3D_PRIMITIVE command.
## Base of Data Stolen Memory

### BDSM - Base of Data Stolen Memory

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>PCI: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0005Ch</td>
</tr>
</tbody>
</table>

This register contains the base address of Graphics Data Stolen DRAM memory. Graphics Stolen Memory is within DRAM space. The base of stolen memory will always be below 4G.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:20</td>
<td>BDSM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BDSM: BASE_OF_Data_STOLEN_MEMORY. This register contains bits 31 to 20 of the base address of Data stolen DRAM memory. For certain GTLC generated accesses, this base register will be added to GTLC-provided offset address, forming the full physical address for the PFI fabric. This is also used as a base for VGA paged accesses.</td>
</tr>
<tr>
<td>19:1</td>
<td>RESERVED</td>
<td>Default Value: 00000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>0</td>
<td>BDSM_LOCK</td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This bit will lock all writable settings in this register, including itself.</td>
</tr>
</tbody>
</table>
Base of GTT Stolen Memory

**BGSM - Base of GTT Stolen Memory**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>PCI: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>00070h</td>
</tr>
</tbody>
</table>

Base of GTT table in Gfx Stolen Memory
The GTT table is located within Graphics Stolen Memory in DRAM space. The base of stolen memory will always be below 4G.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:20</td>
<td><strong>BGSM</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BGSM: Gfx Base of GTT Stolen Memory. This register contains bits 31 to 20 of the base address of GTT Table in stolen DRAM memory. BIOS determines base of GTT stolen memory by subtracting the GTT graphics stolen memory size (PCI offset 50 bits 9:8) from the Graphics Base of Data stolen(PCI offset 5C bits 31:20).</td>
</tr>
<tr>
<td>19:1</td>
<td></td>
<td><strong>RESERVED</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td><strong>BGSM_LOCK</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This bit will lock all writeable settings in this register including itself</td>
</tr>
</tbody>
</table>
## Batch Address Difference Register

<table>
<thead>
<tr>
<th><strong>Register Space</strong></th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Project</strong></td>
<td>CHV, BSW</td>
</tr>
<tr>
<td><strong>Source</strong></td>
<td>PRM</td>
</tr>
<tr>
<td><strong>Default Value</strong></td>
<td>0x00000000</td>
</tr>
<tr>
<td><strong>Access</strong></td>
<td>R/W</td>
</tr>
<tr>
<td><strong>Size (in bits)</strong></td>
<td>32</td>
</tr>
<tr>
<td><strong>Address</strong></td>
<td>02154h-02157h</td>
</tr>
<tr>
<td><strong>Name</strong></td>
<td>Batch Address Difference Register</td>
</tr>
<tr>
<td><strong>ShortName</strong></td>
<td>BB_ADDR_DIFF_RCSUNIT</td>
</tr>
<tr>
<td><strong>Address</strong></td>
<td>12154h-12157h</td>
</tr>
<tr>
<td><strong>Name</strong></td>
<td>Batch Address Difference Register</td>
</tr>
<tr>
<td><strong>ShortName</strong></td>
<td>BB_ADDR_DIFF_VCSUNIT0</td>
</tr>
<tr>
<td><strong>Address</strong></td>
<td>1A154h-1A157h</td>
</tr>
<tr>
<td><strong>Name</strong></td>
<td>Batch Address Difference Register</td>
</tr>
<tr>
<td><strong>ShortName</strong></td>
<td>BB_ADDR_DIFF_VECSUNIT</td>
</tr>
<tr>
<td><strong>Address</strong></td>
<td>1C154h-1C157h</td>
</tr>
<tr>
<td><strong>Name</strong></td>
<td>Batch Address Difference Register</td>
</tr>
<tr>
<td><strong>ShortName</strong></td>
<td>BB_ADDR_DIFF_VCSUNIT1</td>
</tr>
<tr>
<td><strong>Address</strong></td>
<td>22154h-22157h</td>
</tr>
<tr>
<td><strong>Name</strong></td>
<td>Batch Address Difference Register</td>
</tr>
<tr>
<td><strong>ShortName</strong></td>
<td>BB_ADDR_DIFF_BCSUNIT</td>
</tr>
</tbody>
</table>

This register contains the difference between the start of the last batch and where the last initiated Batch Buffer is currently fetching commands.

### Programming Notes

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

<table>
<thead>
<tr>
<th><strong>DWord</strong></th>
<th><strong>Bit</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:2</td>
<td><strong>Batch Buffer Address Difference</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: GraphicsAddress[31:2]</td>
</tr>
</tbody>
</table>

This field specifies the DWord-aligned difference between the starting address of the batch buffer and where the last initiated Batch Buffer is currently fetching commands.

<table>
<thead>
<tr>
<th><strong>DWord</strong></th>
<th><strong>Bit</strong></th>
<th><strong>Reserved</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>1:0</td>
<td>0</td>
<td>MBZ</td>
</tr>
</tbody>
</table>
### Batch Buffer Head Pointer Register

<table>
<thead>
<tr>
<th>BB_ADDR - Batch Buffer Head Pointer Register</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Access: RO</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 02140h-02143h</td>
</tr>
<tr>
<td>Name: Batch Buffer Head Pointer Register</td>
</tr>
<tr>
<td>ShortName: BB_ADDR_RCSUNIT</td>
</tr>
<tr>
<td>Address: 12140h-12143h</td>
</tr>
<tr>
<td>Name: Batch Buffer Head Pointer Register</td>
</tr>
<tr>
<td>ShortName: BB_ADDR_VCSUNIT0</td>
</tr>
<tr>
<td>Address: 1A140h-1A143h</td>
</tr>
<tr>
<td>Name: Batch Buffer Head Pointer Register</td>
</tr>
<tr>
<td>ShortName: BB_ADDR_VECSUNIT</td>
</tr>
<tr>
<td>Address: 1C140h-1C143h</td>
</tr>
<tr>
<td>Name: Batch Buffer Head Pointer Register</td>
</tr>
<tr>
<td>ShortName: BB_ADDR_VCSUNIT1</td>
</tr>
<tr>
<td>Address: 22140h-22143h</td>
</tr>
<tr>
<td>Name: Batch Buffer Head Pointer Register</td>
</tr>
<tr>
<td>ShortName: BB_ADDR_BCSUNIT</td>
</tr>
</tbody>
</table>

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

#### Programming Notes

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:2</td>
<td><strong>Batch Buffer Head Pointer</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: GraphicsAddress[31:2]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This field specifies the DWord-aligned Graphics Memory Address where the last initiated Batch Buffer is currently fetching commands. If no batch buffer is currently active, the Valid bit will be 0 and this field will be meaningless.</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
</tbody>
</table>

Doc Ref # IHD-OS-CHV-BSW-Vol 2c-10.15
### BB_ADDR - Batch Buffer Head Pointer Register

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>Invalid [Default]</td>
<td>Batch buffer Invalid</td>
</tr>
<tr>
<td>1h</td>
<td>Valid</td>
<td>Batch buffer Valid</td>
</tr>
</tbody>
</table>

**Valid**

Format: U1
### Batch Buffer Per Context Pointer

<table>
<thead>
<tr>
<th><strong>Register Space:</strong></th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Project:</strong></td>
<td>CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong></td>
<td>PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong></td>
<td>0x00000000</td>
</tr>
<tr>
<td><strong>Access:</strong></td>
<td>R/W</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong></td>
<td>32</td>
</tr>
<tr>
<td><strong>Trusted Type:</strong></td>
<td>1</td>
</tr>
<tr>
<td><strong>Address:</strong></td>
<td>021C0h-021C3h</td>
</tr>
<tr>
<td><strong>Name:</strong></td>
<td>Batch Buffer Per Context Pointer</td>
</tr>
<tr>
<td><strong>ShortName:</strong></td>
<td>BB_PER_CTX_PTR_RCSUNIT</td>
</tr>
<tr>
<td><strong>Address:</strong></td>
<td>121C0h-121C3h</td>
</tr>
<tr>
<td><strong>Name:</strong></td>
<td>Batch Buffer Per Context Pointer</td>
</tr>
<tr>
<td><strong>ShortName:</strong></td>
<td>BB_PER_CTX_PTR_VCSUNIT0</td>
</tr>
<tr>
<td><strong>Address:</strong></td>
<td>1A1C0h-1A1C3h</td>
</tr>
<tr>
<td><strong>Name:</strong></td>
<td>Batch Buffer Per Context Pointer</td>
</tr>
<tr>
<td><strong>ShortName:</strong></td>
<td>BB_PER_CTX_PTR_VECSUNIT</td>
</tr>
<tr>
<td><strong>Address:</strong></td>
<td>1C1C0h-1C1C3h</td>
</tr>
<tr>
<td><strong>Name:</strong></td>
<td>Batch Buffer Per Context Pointer</td>
</tr>
<tr>
<td><strong>ShortName:</strong></td>
<td>BB_PER_CTX_PTR_VCSUNIT1</td>
</tr>
<tr>
<td><strong>Address:</strong></td>
<td>221C0h-221C3h</td>
</tr>
<tr>
<td><strong>Name:</strong></td>
<td>Batch Buffer Per Context Pointer</td>
</tr>
<tr>
<td><strong>ShortName:</strong></td>
<td>BB_PER_CTX_PTR_BCSUNIT</td>
</tr>
</tbody>
</table>

This register is used to program the batch buffer address to be executed between context restore and execution of ring/execution list if enabled. This will only get executed due to regular context save/restore and not during power restore. This register is part of the execution list context and will be executed per context. Only supported if execution list is enabled. There is no preempting workloads within the Per Context Batch Buffer.

<table>
<thead>
<tr>
<th><strong>Source</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>BlitterCS, VideoCS, VideoCS2, VideoEnhancementCS</td>
</tr>
<tr>
<td>RenderCS</td>
</tr>
<tr>
<td>RenderCS</td>
</tr>
<tr>
<td>RenderCS</td>
</tr>
</tbody>
</table>
### BB_PER_CTX_PTR - Batch Buffer Per Context Pointer

#### Workaround

**Workaround:**
[Render CS Only][Execlist Mode of Scheduling]: SW must ensure arbitration is switched off while context restore is in progress for any given context. This is achieved by disabling arbitration by programming MI_ARB_ON_OFF to “Arbitration Disable” in RCS_INDIRECT_CTX buffer and by enabling back the arbitration by programming MI_ARB_ON_OFF to “Arbitration Enable” as the last command prior to MI_BATCH_END in the BB_PER_CTX_PTR buffer of every context submitted. Note that RCS_INDIRECT_CTX_OFFSET could be set to default value or any other legitimate value as per the programming notes of the register definition. Arbitration disable by programming MI_ARB_ON_OFF (Arbitration Disabled) in RCS_INDIRECT_CTX buffer. Arbitration enabled by programming MI_ARB_ON_OFF (Arbitration Enable) as the last command prior to MI_BATCH_BUFFER_END in BB_PER_CTX_PTR buffer. Additional Note: This WA need not be applied when it is guaranteed for no preemption to occur during execution of GPGPU workload. Preemption of GPGPU workload can be avoided by Bracketing the GPGPU workload with MI_ARB_ON_OFF (Arbitration Disable) and MI_ARB_ON_OFF (Arbitration Enable) command. MI_ARB_ON_OFF is a privileged command and can only be programmed in ring buffer or in privileged batch buffer (batch buffer in GGTT memory). Pending execlist submitted must not trigger preemption of the ongoing GPGPU workload due to following reasons First context of the pending execlist submitted is not the same as the ongoing GPGPU context. Force restore bit set for the submitted pending execlist.

**Workaround:**

To work around a known HW issue, SW must do the below Programming Sequence prior to programming MI_BATCH_BUFFER_END command in BB_PER_CTX_PTR. SW must ensure both MI_LOAD_REGISTER_REG and MI_BATCH_BUFFER_END commands mentioned in the below sequence are placed in the same cacheline of memory.

1. **MI_LOAD_REGISTER_IMM:** 0x00800000 à 0x20C0
2. **MI_ATOMIC**
   - a. Set “CS STALL” (Dword0[17])
   - b. “Return Data Control” enabled (Dword0[16])
   - c. “ATOMIC OPCODE” set to LOAD operation (Dword0[15:8]= 0x4)
   - d. “Memory Address” set to scratch space in GFX memory.
   - e. “Operand1 Data Dword 0” must be programmed to 0x0080_0080
3. **MI_LOAD_REGISTER_MEM**
   - a. Set “Async Mode Enable” (Dword0[21])
   - b. “Memory Address” set to same as in MI_ATOMIC command above.
   - c. “Register Address” set to 0x20C0
4. **MI_LOAD_REGISTER_REG:** 0x215C à 0x215C
5. **MI_BATCH_BUFFER_END** // Note that there shouldn’t be any commands programmed between step4 & step5 and also these commands must be placed in the same cacheline of memory.

#### Additional Note:
This workaround need not be applied when Resource Streamer (RS) is not enabled or when a Resource Streamer enabled context is guaranteed not to be preempted.

- Preemption of RS enabled workload can be avoided by
  - Bracketing the RS enabled workload with MI_ARB_ON_OFF (Arbitration Disable) and MI_ARB_ON_OFF (Arbitration Enable) command. MI_ARB_ON_OFF is a privileged command and can only be programmed in ring buffer or in privileged batch buffer (batch buffer in GGTT memory).
  - Pending execlist submitted must not trigger preemption of the ongoing RS enabled workload due to following reasons
    - First context of the pending execlist submitted is not the same as the ongoing RS enabled context.
    - Force restore bit set for the submitted pending execlist.

### BB_PER_CTX_PTR - Batch Buffer Per Context Pointer

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:12</td>
<td><strong>Batch Buffer Per Context Address</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Pointer to the Context in memory to be executed as a batch.</td>
</tr>
<tr>
<td></td>
<td>11:2</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td><strong>RS Enabled Batch Buffer Per Context</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: U1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If set, the command stream will enable the RS to parse commands.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Programming Notes</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>This must be set when programming the resource streamer pool commands (3DSTATE_BINDING_TABLE_POOL_ALLOC, 3DSTATE_GATHER_POOL_ALLOC, and 3DSTATE_DX9_CONSTANT_BUFFER_POOL_ALLOC) in order for the pool alloc fields to be valid in both the render engine and resource streamer.</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td><strong>Batch Buffer Per Context Valid</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: U1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If set, the command stream will execute the context from the <strong>Batch Buffer Per Context Address</strong> prior to the execution of actual submitted workloads.</td>
</tr>
</tbody>
</table>

**Programming Notes**

This must be set when programming the resource streamer pool commands (3DSTATE_BINDING_TABLE_POOL_ALLOC, 3DSTATE_GATHER_POOL_ALLOC, and 3DSTATE_DX9_CONSTANT_BUFFER_POOL_ALLOC) in order for the pool alloc fields to be valid in both the render engine and resource streamer.
## Batch Buffer Start Head Pointer Register

<table>
<thead>
<tr>
<th>BB_START_ADDR - Batch Buffer Start Head Pointer Register</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> RenderCS</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Address:</strong></th>
<th>02150h</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Address:</strong></td>
<td>12150h-12153h</td>
</tr>
<tr>
<td><strong>Name:</strong></td>
<td>Batch Buffer Start Head Pointer Register</td>
</tr>
<tr>
<td><strong>ShortName:</strong></td>
<td>BB_START_ADDR_VCSUNIT0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Address:</strong></th>
<th>1A150h-1A153h</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name:</strong></td>
<td>Batch Buffer Start Head Pointer Register</td>
</tr>
<tr>
<td><strong>ShortName:</strong></td>
<td>BB_START_ADDR_VECSUNIT</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Address:</strong></th>
<th>1C150h-1C153h</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name:</strong></td>
<td>Batch Buffer Start Head Pointer Register</td>
</tr>
<tr>
<td><strong>ShortName:</strong></td>
<td>BB_START_ADDR_VCSUNIT1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Address:</strong></th>
<th>22150h-22153h</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name:</strong></td>
<td>Batch Buffer Start Head Pointer Register</td>
</tr>
<tr>
<td><strong>ShortName:</strong></td>
<td>BB_START_ADDR_BCSUNIT</td>
</tr>
</tbody>
</table>

This register contains the address specified in the last MI_START_BATCH_BUFFER command.

### Programming Notes

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:2</td>
<td><strong>Batch Buffer Start Head Pointer</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format:</strong> GraphicsAddress[31:2]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This field specifies the DWord-aligned Graphics Memory Address where the last initiated Batch Buffer starting address.</td>
</tr>
</tbody>
</table>

<p>| 1     |     | <strong>Preempted Batch Buffer RS Control Stop Flag</strong>  |
|       |     | <strong>Format:</strong> Flag  |
|       |     | This field specifies RS Control Stop Flag when a batch buffer is preempted. This is for HW internal use and should not be written by SW. This bit gets reset when RS_PREEMPTED field of RS_PREEMPT_STATUS is written Zero. |</p>
<table>
<thead>
<tr>
<th>BB_START_ADDR - Batch Buffer Start Head Pointer Register</th>
</tr>
</thead>
<tbody>
<tr>
<td>This bit is set by:</td>
</tr>
<tr>
<td>- Ctx restore of this bit</td>
</tr>
<tr>
<td>- MI_RS_CONTROL_STOP (except for the ctx restore command)</td>
</tr>
<tr>
<td>This bit is cleared by:</td>
</tr>
<tr>
<td>- MI_RS_CONTROL_START</td>
</tr>
<tr>
<td>- Any Batch start except resubmitted RS batch</td>
</tr>
<tr>
<td>- A batch end that doesn't include preemption</td>
</tr>
<tr>
<td>- Ctx save</td>
</tr>
<tr>
<td>Writing 0 to bit[0] of the RS STATUS register</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
Batch Buffer Start Head Pointer Register for Upper DWord

**BB_START_ADDR_UDW - Batch Buffer Start Head Pointer Register for Upper DWord**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>RenderCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>02170h</td>
</tr>
</tbody>
</table>

**Address:** 12170h-12173h  
**Name:** Batch Buffer Start Upper Head Pointer Register  
**ShortName:** BB_START_ADDR_UDW_VCSUNIT0

**Address:** 1A170h-1A173h  
**Name:** Batch Buffer Start Upper Head Pointer Register  
**ShortName:** BB_START_ADDR_UDW_VECSUNIT

**Address:** 1C170h-1C173h  
**Name:** Batch Buffer Start Upper Head Pointer Register  
**ShortName:** BB_START_ADDR_UDW_VCSUNIT1

**Address:** 22170h-22173h  
**Name:** Batch Buffer Start Upper Head Pointer Register  
**ShortName:** BB_START_ADDR_UDW_BCSUNIT

This register contains the 4GB aligned base address of gfx 4GB virtual address space within the host's 64-bit virtual address space specified in the last MI_START_BATCH_BUFFER command.

**Programming Notes**

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td>Format:</td>
<td>MBZ</td>
</tr>
<tr>
<td>15:0</td>
<td></td>
<td><strong>Batch Buffer Start Head Pointer Upper DWORD</strong></td>
</tr>
<tr>
<td></td>
<td>Format:</td>
<td>GraphicsAddress[47:32]</td>
</tr>
</tbody>
</table>

This field specifies the 4GB aligned base address of gfx 4GB virtual address space within the host's 64-bit virtual address space for the last initiated Batch Buffer starting address.
Batch Buffer State Register

<table>
<thead>
<tr>
<th>BB_STATE - Batch Buffer State Register</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000 CHV, BSW</td>
</tr>
<tr>
<td>Access: RO</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 02110h-02113h</td>
</tr>
<tr>
<td>Name: Batch Buffer State Register</td>
</tr>
<tr>
<td>ShortName: BB_STATE_RCSUNIT</td>
</tr>
<tr>
<td>Address: 12110h-12113h</td>
</tr>
<tr>
<td>Name: Batch Buffer State Register</td>
</tr>
<tr>
<td>ShortName: BB_STATE_VCSUNIT0</td>
</tr>
<tr>
<td>Address: 1A110h-1A113h</td>
</tr>
<tr>
<td>Name: Batch Buffer State Register</td>
</tr>
<tr>
<td>ShortName: BB_STATE_VECSUNIT</td>
</tr>
<tr>
<td>Address: 1C110h-1C113h</td>
</tr>
<tr>
<td>Name: Batch Buffer State Register</td>
</tr>
<tr>
<td>ShortName: BB_STATE_VCSUNIT1</td>
</tr>
<tr>
<td>Address: 22110h-22113h</td>
</tr>
<tr>
<td>Name: Batch Buffer State Register</td>
</tr>
<tr>
<td>ShortName: BB_STATE_BCSUNIT</td>
</tr>
</tbody>
</table>

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

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:8</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: All</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: All</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Source: BlitterCS, VideoCS, VideoCS2, VideoEnhancementCS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
</tbody>
</table>
## BB_STATE - Batch Buffer State Register

<table>
<thead>
<tr>
<th>7</th>
<th>Resource Streamer Enable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>RenderCS</td>
</tr>
<tr>
<td>Format:</td>
<td>U1</td>
</tr>
</tbody>
</table>

When this bit is set, the Resource Streamer will execute the batch buffer. When this bit is clear the Resource Streamer will not execute the batch buffer.

| 6 | Reserved |

<table>
<thead>
<tr>
<th>6</th>
<th>2nd Level Buffer Security Indicator</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>VideoCS, VideoCS2</td>
</tr>
<tr>
<td>Exists If:</td>
<td>//VCS, VCS2</td>
</tr>
</tbody>
</table>

If set, VCS is fetching 2nd level batch commands from a PPGTT address space. If clear, GGTT. If execlists are enabled and this is set, the batch buffer is non-secure and cannot execute privileged commands nor access privileged (GGTT) memory. It will be accessed via the PPGTT. If clear, this batch buffer is secure and will be accessed via the GGTT. Note: When execlists are enabled this field reflects the effective security level and may not be the same as the Buffer Security Indicator written using MI_BATCH_BUFFER_START.

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>MIBUFFER_SECURE [Default]</td>
<td>Located in GGTT memory</td>
</tr>
<tr>
<td>1h</td>
<td>MIBUFFER_NONSECURE</td>
<td>Located in PPGTT memory</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>6</th>
<th>2nd Level Buffer Security Indicator</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>BlitterCS, VideoEnhancementCS</td>
</tr>
<tr>
<td>Exists If:</td>
<td>//BCS, VECS</td>
</tr>
<tr>
<td>Format:</td>
<td>MI_2ndBufferSecurityType</td>
</tr>
</tbody>
</table>

If set, VECS is fetching 2nd level batch commands from a PPGTT address space. If clear, GGTT. If execlists are enabled and this is set, the batch buffer is non-secure and cannot execute privileged commands nor access privileged (GGTT) memory. It will be accessed via the PPGTT. If clear, this batch buffer is secure and will be accessed via the GGTT.

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>MIBUFFER_SECURE [Default]</td>
<td>Located in GGTT memory</td>
</tr>
<tr>
<td>1h</td>
<td>MIBUFFER_NONSECURE</td>
<td>Located in PPGTT memory</td>
</tr>
</tbody>
</table>

### Programming Notes

When execlists are enabled, this field reflects the effective security level and may not be the same as the Buffer Security Indicator written using MI_BATCH_BUFFER_START.
### BB_STATE - Batch Buffer State Register

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>GGTT [Default]</td>
<td>This Batch buffer is located in GGTT memory and is privileged</td>
</tr>
<tr>
<td>1h</td>
<td>PPGTT</td>
<td>This Batch buffer is located in PPGTT memory and is non-privileged.</td>
</tr>
</tbody>
</table>

#### Address Space Indicator

- **Project:** CHV, BSW

Note: 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.

- **Value 4 Reserved**
  - **Project:** All
  - **Source:** BlitterCS
  - **Exists If:** //BCS
  - **Format:** MBZ

- **Value 3:0 Reserved**
  - **Project:** All
  - **Format:** MBZ
Batch Buffer Upper Head Pointer Preemption Register

**BB_PREEMPT_ADDR_UDW - Batch Buffer Upper Head Pointer Preemption Register**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
</tbody>
</table>

| Address:        | 0216Ch-0216Fh |
| Name:           | Batch Buffer Upper Head Pointer Preemption Register |
| ShortName:      | BB_PREEMPT_ADDR_UDW_RCSUNIT |

| Address:        | 1216Ch-1216Fh |
| Name:           | Batch Buffer Upper Head Pointer Preemption Register |
| ShortName:      | BB_PREEMPT_ADDR_UDW_VCSUNIT0 |

| Address:        | 1A16Ch-1A16Fh |
| Name:           | Batch Buffer Upper Head Pointer Preemption Register |
| ShortName:      | BB_PREEMPT_ADDR_UDW_VECSUNIT |

| Address:        | 1C16Ch-1C16Fh |
| Name:           | Batch Buffer Upper Head Pointer Preemption Register |
| ShortName:      | BB_PREEMPT_ADDR_UDW_VCSUNIT1 |

| Address:        | 2216Ch-2216Fh |
| Name:           | Batch Buffer Upper Head Pointer Preemption Register |
| ShortName:      | BB_PREEMPT_ADDR_UDW_BCSUNIT |

This register contains the 4GB aligned base address of gfx 4GB virtual address space within the host’s 64-bit virtual address space of the last preempted batch buffer. This register follows the same rules as the BB_PREEMPT_ADDR register.

---

### Programming Notes

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
<tr>
<td>15:0</td>
<td></td>
<td><strong>Batch Buffer Head Pointer Upper DWORD</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: GraphicsAddress[47:32]</td>
</tr>
</tbody>
</table>

This field specifies the 4GB aligned base address of gfx 4GB virtual address space within the host’s 64-bit virtual address space of the last preempted batch buffer.
### Batch Buffer Upper Head Pointer Register

**Register Space:** MMIO: 0/2/0  
**Project:** CHV, BSW  
**Source:** PRM  
**Default Value:** 0x00000000  
**Access:** RO  
**Size (in bits):** 32

<table>
<thead>
<tr>
<th>Address</th>
<th>Name:</th>
<th>ShortName:</th>
</tr>
</thead>
<tbody>
<tr>
<td>02168h-0216Bh</td>
<td>Batch Buffer Upper Head Pointer Register</td>
<td>BB_ADDR_UDW_RCSUNIT</td>
</tr>
<tr>
<td>12168h-1216Bh</td>
<td>Batch Buffer Upper Head Pointer Register</td>
<td>BB_ADDR_UDW_VCSUNIT0</td>
</tr>
<tr>
<td>1A168h-1A16Bh</td>
<td>Batch Buffer Upper Head Pointer Register</td>
<td>BB_ADDR_UDW_VECSUNIT</td>
</tr>
<tr>
<td>1C168h-1C16Bh</td>
<td>Batch Buffer Upper Head Pointer Register</td>
<td>BB_ADDR_UDW_VCSUNIT1</td>
</tr>
<tr>
<td>22168h-2216Bh</td>
<td>Batch Buffer Upper Head Pointer Register</td>
<td>BB_ADDR_UDW_BCSUNIT</td>
</tr>
</tbody>
</table>

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

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
</table>
| 0     | 31:16| **Reserved**  
Format: MBZ |
| 15:0  |      | **Batch Buffer Head Pointer Upper DWord**  
Format: GraphicsAddress[47:32] |

This field specifies the 4GB aligned base address of gfx 4GB virtual address space within the host’s 64-bit virtual address space where the last initiated Batch Buffer is currently fetching commands. If no batch buffer is currently active, the Valid bit in BB_ADDR will be 0 and this field is meaningless.
**BCS_PREEMPTION_HINT**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>BlitterCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>224BCh</td>
</tr>
</tbody>
</table>

This register contains the Head pointer offset in to the Ring Buffer or the Dword aligned Graphics address in to the Batch Buffer corresponding to either MI_ARB_CHECK called Preemption Hint Address. When Preemption Hint Address is enabled, BCS will honor UHPTR only on parsing MI_ARB_CHK at Preemption Hint Address.

This register contains the Head pointer offset into the Ring Buffer or the Dword aligned Graphics address into the Batch Buffer corresponding to the below listed commands in execlist mode of operation

- MI_ARB_CHECK
- MI_WAIT_FOR_EVENT
- MI_SEMAPHORE_WAIT

**Programming Notes**

**Programming Restriction:** This register should NEVER be programmed in functional mode, this should be used only in validation mode to achieve deterministic behavior of UHPTR being sampled by a given MI_ARB_CHECK in command stream. Programmer has to ensure that BCS Preemption Hint register gets programmed before UHPTR is programmed and well before BCS crosses the corresponding execution point. Preemption hint for both RingBuffer and Batch Buffer can't be enabled simultaneously.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:2</td>
<td>Preempted Hint Address</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: U30</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: GraphicsAddress[31:2]</td>
</tr>
</tbody>
</table>

This field contains the Head offset in to the Ring Buffer when Preemption Hint is set to Ring Buffer and Dword aligned Graphics Address in to the batch buffer when Preemption Hint is set to Batch Buffer.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>Batch Buffer Preemption Hint</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: Enable</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>Disabled</td>
<td>Preemption hint is disabled in batch buffer.</td>
</tr>
<tr>
<td>1h</td>
<td>Enabled</td>
<td>Preemption hint is enabled in ring buffer and preemption hint address corresponds to the instruction in Batch Buffer.</td>
</tr>
</tbody>
</table>
## BCS_PREEMPTION_HINT - BCS_PREEMPTION_HINT

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>Disable</td>
<td>Preemption hint is disabled in ring buffer.</td>
</tr>
<tr>
<td>1h</td>
<td>Enabled</td>
<td>Preemption hint is enabled in ring buffer and preemption hint address</td>
</tr>
<tr>
<td></td>
<td></td>
<td>corresponds to the instruction in Ring Buffer.</td>
</tr>
</tbody>
</table>

**Ring Preemption Hint**

Format: Enable
BCS_PREEMPTION_HINT_UDW

<table>
<thead>
<tr>
<th>BCS_PREEMPTION_HINT_UDW - BCS_PREEMPTION_HINT_UDW</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: BlitterCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Access: R/W</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 224C8h</td>
</tr>
</tbody>
</table>

This register contains the 4GB aligned base address of gfx 4GB virtual address space within the host's 64-bit virtual address space of the Batch Buffer corresponding to MI_ARB_CHECK command called Preemption Hint Address.

### Programming Notes

**Programming Restriction:** This register should NEVER be programmed in functional mode, this should be used only in validation mode to achieve deterministic behavior of UH PTR being sampled by a given MI_ARB_CHK in command stream.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
<tr>
<td>15:0</td>
<td></td>
<td><strong>Preempted Hint Address Upper DWORD</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: GraphicsAddress[47:32]</td>
</tr>
</tbody>
</table>

This field contains the 4GB aligned base address of gfx 4GB virtual address space within the host's 64-bit virtual address space of the batch buffer when Preemption Hint is set to Batch Buffer. This field is not valid when Preemption Hint is set to Ring Buffer.
## BCS Active Upper Head Pointer Register

<table>
<thead>
<tr>
<th>BCS_ACTHD_UDW - BCS Active Upper Head Pointer Register</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: BlitterCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Access: RO</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 2205Ch</td>
</tr>
</tbody>
</table>

This register contains the Head "Pointer" (4GB aligned base address of gfx 4GB virtual address space within the host's 64-bit virtual address space) of the currently-active batch buffer.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
<tr>
<td>15:0</td>
<td></td>
<td><strong>Head Pointer Upper DWORD</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: GraphicsAddress[47:32]</td>
</tr>
</tbody>
</table>

4GB aligned base address of gfx 4GB virtual address space within the host's 64-bit virtual address space corresponding to the Head Pointer of the currently-active batch buffer.
BCS Context ID Preemption Hint

**BCS_CTXID_PREEMPTION_HINT - BCS Context ID Preemption Hint**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>BlitterCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>224CCh</td>
</tr>
</tbody>
</table>

This register contains the Context ID of a context in execlist mode of operation. In execlist mode of operation BCS_PREEMPTION_HINT registers are looked at by Blitter Command Streamer on executing a context having Context ID that matches with the contents of this register. This register contents are valid and looked at only in execlist mode of operation.

**Programming Notes**

This register should NEVER be programmed in functional mode, this must be used only in validation mode to achieve deterministic preemption behavior in execlist mode of operation.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Context ID Preemption Hint</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Format:</th>
<th>U32</th>
</tr>
</thead>
</table>

If 0 this field has no effect. If nonzero it indicates the only context ID that can be preempted when execlists are enabled. A preemption attempt when the context ID of the currently executing ring context does not match this field will be ignored.
# BCS Context Sizes

<table>
<thead>
<tr>
<th>BCS_CXT_SIZE - BCS Context Sizes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> BlitterCS</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000A05 CHV, BSW</td>
</tr>
<tr>
<td><strong>Access:</strong> Read/32 bit Write Only</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 221A8h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:13</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: All</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
<tr>
<td>12:8</td>
<td></td>
<td><strong>BCS Context Size</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: All</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: U5</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Value</strong></td>
</tr>
<tr>
<td>Ah</td>
<td></td>
<td>[Default]</td>
</tr>
<tr>
<td>7:5</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: All</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
<tr>
<td>4:0</td>
<td></td>
<td><strong>Execlist Context Size</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: U5</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Value</strong></td>
</tr>
<tr>
<td>5h</td>
<td></td>
<td>[Default]</td>
</tr>
</tbody>
</table>
# BCS Context Timestamp Count

<table>
<thead>
<tr>
<th>BCS_CTX_TIMESTAMP - BCS Context Timestamp Count</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> BlitterCS</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
</tbody>
</table>

**Address:** 0x223A8h

This register provides a mechanism to obtain cumulative run time of a GPU context on HW. This register gets context save/restored on a context switch. SW must reset this register on very first submission of a context to HW, then afterwards gets context save/restored maintaining the cumulative run time of the corresponding context. Note that the value of this register can be obtained in a 3D pipeline-synchronous fashion without a pipeline flush by using the PIPE_CONTROL command. See 3D Geometry Pipeline in the "3D and Media" volume. This register (effectively) counts at a constant frequency by adjusting the increment amount according to the actual reference clock frequency. SW therefore does not need to know the reference clock frequency. This register is context save restore on a context switch.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Timestamp Value</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format:</strong> U32</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>This register increments for every 80 ns of time.</strong></td>
</tr>
</tbody>
</table>
# BCS Counter for the Blitter Engine

<table>
<thead>
<tr>
<th>BCS_CNTR - BCS Counter for the Blitter Engine</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: BlitterCS</td>
</tr>
<tr>
<td>Default Value: 0xFFFFFFFFFF</td>
</tr>
<tr>
<td>Access: R/W</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 22178h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Count Value</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: ffffffffh</td>
</tr>
</tbody>
</table>

Writing a Zero value to this register starts the counting. Writing a Value of FFFF FFFF to this counter stops the counter.
BCS Error Identity Register

### BCS_EIR - BCS Error Identity Register

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>BlitterCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/WC</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>220B0h</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Project:</strong> All</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format:</strong> MBZ</td>
</tr>
<tr>
<td>0</td>
<td>15:0</td>
<td>Error Identity Bits</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Project:</strong> All</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format:</strong> Array of Error condition bits See Table 1.5. Hardware-Detected Error Bits</td>
</tr>
</tbody>
</table>

This register contains the persistent values of ESR error status bits that are unmasked via the EMR register. The logical OR of all (defined) bits in this register is reported in the Master Error bit of the Interrupt Status Register. To clear an error condition, software must first clear the error by writing a '1' to the appropriate bit(s) in this field. If required, software should then proceed to clear the Master Error bit of the IIR.

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>[Default]</td>
<td></td>
</tr>
<tr>
<td>1h</td>
<td>Error occurred</td>
<td>Error occurred</td>
</tr>
</tbody>
</table>

### Programming Notes

Writing a '1' to a set bit will cause that error condition to be cleared. However, the Instruction Error bit (Bit 0) cannot be cleared except by reset (i.e., it is a fatal error).
## BCS Error Mask Register

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FFFFh</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Must Be One</td>
</tr>
<tr>
<td>15:0</td>
<td></td>
<td><strong>Error Mask Bits</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>All</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Array of error condition mask bits See Table 15. Hardware-Detected Error Bits</td>
</tr>
</tbody>
</table>

This register contains a bit mask that selects which error condition bits (from the ESR) are reported in the EIR.

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000h</td>
<td>Not Masked</td>
<td>Will be reported in the EIR</td>
</tr>
<tr>
<td>FFFFh</td>
<td>Masked [Default]</td>
<td>Will not be reported in the EIR</td>
</tr>
</tbody>
</table>
# BCS Error Status Register

**BCS_ESR - BCS Error Status Register**

<table>
<thead>
<tr>
<th>Description</th>
<th>DWord</th>
<th>Bit</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Reserved</strong></td>
<td>0</td>
<td>31:16</td>
<td><strong>Error Status Bits</strong></td>
</tr>
<tr>
<td>Project:</td>
<td></td>
<td></td>
<td>All</td>
</tr>
<tr>
<td>Format:</td>
<td></td>
<td></td>
<td>MBZ</td>
</tr>
<tr>
<td><strong>Error Status Bits</strong></td>
<td>15:0</td>
<td></td>
<td>Array of error condition bits See Table 15. Hardware-Detected Error Bits</td>
</tr>
<tr>
<td>Project:</td>
<td></td>
<td></td>
<td>All</td>
</tr>
<tr>
<td>Format:</td>
<td></td>
<td></td>
<td>Array of error condition bits See Table 15. Hardware-Detected Error Bits</td>
</tr>
</tbody>
</table>

This register contains the non-persistent values of all hardware-detected error condition bits.

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>[Default]</td>
<td></td>
</tr>
<tr>
<td>1h</td>
<td>Error Condition Detected</td>
<td>Error Condition detected</td>
</tr>
</tbody>
</table>
## BCS Execute Condition Code Register

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
</table>
| 0     | 31:16 | **Mask Bits**  
|       | Format: | Mask[15:0]  
|       | These bits serves as a write enable for bits 15:0. If this register is written with any of these bits clear the corresponding bit in the field 15:0 will not be modified. Reading these bits always returns 0s. |
| 15    |       | **Reserved**  
|       | Format: | MBZ  
| 14    |       | **Context Wait for V-blank on Pipe-C**  
|       | Project: | CHV, BSW  
|       | This field when set indicates the corresponding context has executed MI_WAIT_FOR_EVENT with "Display Pipe C Vertical Blank Wait Enable" set. This is an internal HW flag and should not be accessed by SW. |
| 13    |       | **Context Wait for V-blank on Pipe-B**  
|       | Project: | CHV, BSW  
|       | This field when set indicates the corresponding context has executed MI_WAIT_FOR_EVENT with "Display Pipe B Vertical Blank Wait Enable" set. This is an internal HW flag and should not be accessed by SW. |
| 12    |       | **Context Wait for V-blank on Pipe-A**  
|       | Project: | CHV, BSW  
<p>|       | This field when set indicates the corresponding context has executed MI_WAIT_FOR_EVENT with &quot;Display Pipe A Vertical Blank Wait Enable&quot; set. This is an internal HW flag and should not be accessed by SW. |</p>
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Format</th>
<th>Project</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>11:5</td>
<td>Reserved</td>
<td>MBZ</td>
<td>CHV, BSW</td>
<td>MBZ</td>
</tr>
<tr>
<td>4:0</td>
<td>Reserved</td>
<td>MBZ</td>
<td>CHV, BSW</td>
<td>MBZ</td>
</tr>
</tbody>
</table>

**BCS_EXCC - BCS Execute Condition Code Register**
BCS General Purpose Register

**BCS_GPR - BCS General Purpose Register**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>BlitterCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000, 0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>64</td>
</tr>
<tr>
<td>Address:</td>
<td>22600h-2267Fh</td>
</tr>
</tbody>
</table>

This is a general purpose register bank of sixteen 64-bit registers, used as temporary storage by the MI_MATH command to do ALU operations.

**Programming Notes**

Any operation that initiates a read to register 0x2266C will return the value of 0x2260c register. This does not include context save or MI_MATH command operation.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>63:0</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
</tbody>
</table>
BCS Hardware Status Mask Register

<table>
<thead>
<tr>
<th>BCS_HWSTAM - BCS Hardware Status Mask Register</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: BlitterCS</td>
</tr>
<tr>
<td>Default Value: 0xFFFFFFFF</td>
</tr>
<tr>
<td>Access: R/W</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Trusted Type: 1</td>
</tr>
<tr>
<td>Address: 22098h</td>
</tr>
</tbody>
</table>

Access: RO for Reserved Control bits

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

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Hardware Status Mask Register</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: FFFFFFFFh</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: All</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: Array of Masks</td>
</tr>
<tr>
<td></td>
<td></td>
<td>refer to Table 5-1 in Interrupt Control Register section for bit definitions</td>
</tr>
</tbody>
</table>
BCS IDLE Max Count

**BCS_PWRCTX_MAXCNT - BCS IDLE Max Count**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>BlitterCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000040 CHV, BSW</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Trusted Type:</td>
<td>1</td>
</tr>
<tr>
<td>Address:</td>
<td>22054h</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:20</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Project:</strong> All</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format:</strong> MBZ</td>
</tr>
<tr>
<td>19:0</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
</tbody>
</table>
# BCS Idle Switch Delay

## BCS_IDLEDLY - BCS Idle Switch Delay

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>BlitterCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>2223Ch</td>
</tr>
</tbody>
</table>

The IDLEDLY register contains an Idle Delay field which specifies the minimum number of microseconds allowed for command streamer to wait before a context is switched out leading to IDLE state in Exclist mode, i.e following this context switch there is no active element available in HW to execute.

A default value of 0, means that by default, there is no restriction to wait on a context switch leading to IDLE. This register has no significance when exectls are not enabled.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Project:</th>
<th>Format:</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:21</td>
<td><strong>Reserved</strong></td>
<td>All</td>
<td>MBZ</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>IDLE Delay</strong></td>
<td>All</td>
<td>U21</td>
</tr>
<tr>
<td></td>
<td>20:0</td>
<td>Minimum number of micro-seconds allowed</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## BCS Instruction Parser Mode Register

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td><strong>Mask Bits</strong></td>
</tr>
<tr>
<td></td>
<td>Format:</td>
<td>Mask[15:0]</td>
</tr>
<tr>
<td></td>
<td>Must be set to modify corresponding bit in Bits 15:0. (All implemented bits)</td>
<td></td>
</tr>
</tbody>
</table>

| 15:11 | **Reserved** |
|       | Project: | All |
|       | Format: | MBZ |

| 10     | **Reserved** |

| 9      | **Reserved** |
|       | Project: | CHV, BSW |
|       | Format: | MBZ |

| 8:7    | **Reserved** |
|       | Project: | All |
|       | Format: | MBZ |

| 6:5    | **Reserved** |
|       | Project: | CHV, BSW |
|       | Format: | MBZ |

| 4:0    | **Reserved** |
|       | Project: | All |
|       | Format: | MBZ |
## BCS Interrupt Mask Register

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Interrupt Mask Bits</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: All</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: Array of interrupt mask bits Refer to Table 5-1 in Interrupt Control Register section for bit definitions</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This field contains a bit mask which selects which interrupt bits (from the ISR) are reported in the IIR.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FFFF FFFFh</td>
<td>[Default]</td>
<td></td>
</tr>
<tr>
<td>0h</td>
<td>Not Masked</td>
<td>Will be reported in the IIR</td>
</tr>
<tr>
<td>1h</td>
<td>Masked</td>
<td>Will not be reported in the IIR</td>
</tr>
</tbody>
</table>
### BCS Mode Register for Software Interface

**BCS_MI_MODE - BCS Mode Register for Software Interface**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>BlitterCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000200 CHV, BSW</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>2209Ch-2209Fh</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td>Masks</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A 1 in a bit in this field allows the modification of the corresponding bit in Bits 15:0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>15</td>
<td>Suspend Flush</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: All</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Mask: MMIO(0x209c)#31</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>No Delay</td>
<td>HW will not delay flush, this bit will get cleared by MI_SUSPEND_FLUSH as well</td>
</tr>
<tr>
<td>1h</td>
<td>Delay Flush</td>
<td>Suspend flush is active</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>14:12</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Read/Write</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>11</td>
<td>Invalidate UHPTR enable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If bit set H/W clears the valid bit of BCS_UHPTR (4134h, bit 0) when current active head pointer is equal to UHPTR.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10</td>
<td>Atomic Read Return for MI_COPY_MEM_MEM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>Disable [Default]</td>
<td>Hardware does a regular memory fence write to complete the write to the destination address before moving to the next instruction.</td>
</tr>
<tr>
<td>1h</td>
<td>Enable</td>
<td>Hardware does Atomic Move with Read Return to complete the write to the destination address before moving to the next instruction.</td>
</tr>
</tbody>
</table>
## BCS_MI_MODE - BCS Mode Register for Software Interface

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Value</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>Ring Idle (Read Only Status Bit)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><em>W</em>rites to this bit are not allowed.*</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>Parser not idle</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>Parser idle [Default]</td>
</tr>
<tr>
<td>8</td>
<td>Stop Ring</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Software must set this bit to force the Ring and Command Parser to Idle. Software must read a 1 in Ring Idle bit after setting this bit to ensure that the hardware is idle. <em>Software must clear this bit for Ring to resume normal operation.</em></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>Normal Operation [Default]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>Parser is turned off</td>
</tr>
<tr>
<td>7:2</td>
<td>Reserved</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Read/Write</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Bypass Fence Write</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>If set, this bit will bypass all writes during flushes, independent of programming. This includes post-sync op bits, the implicit TLB invalidate write (set in GFX_MODE[13]), and sync flush fences. <em>Note this is only intended for work-arounds</em></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>Normal Operation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>Bypass</td>
</tr>
<tr>
<td>0</td>
<td>Reserved</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Read/Write</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# BCS PPGTT Directory Cacheline Valid Register

<table>
<thead>
<tr>
<th>BCS_PP_DCLV - BCS PPGTT Directory Cacheline Valid Register</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: BitterCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000, 0x00000000</td>
</tr>
<tr>
<td>Access: R/W</td>
</tr>
<tr>
<td>Size (in bits): 64</td>
</tr>
<tr>
<td>Address: 22220h</td>
</tr>
</tbody>
</table>

Default Value = 0h

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

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

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>63:32</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
<tr>
<td>31:0</td>
<td>16 entries</td>
<td><strong>PPGTT Directory Cache Restore</strong></td>
</tr>
<tr>
<td></td>
<td>Format:</td>
<td>Enable[32]</td>
</tr>
<tr>
<td></td>
<td>[1..32]</td>
<td>16 entries</td>
</tr>
<tr>
<td></td>
<td>If set, the [1st..32nd] 16 entries of the directory cache are considered valid and will be brought in on context restore. If clear, these entries are considered invalid and fetch of these entries will not be attempted.</td>
<td></td>
</tr>
</tbody>
</table>
**BCS Primary DMA Engine Fetch Upper Address**

<table>
<thead>
<tr>
<th>BCS_DMA_FADD_P_UDW - BCS Primary DMA Engine Fetch Upper Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: BlitterCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Access: RO</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 22060h</td>
</tr>
</tbody>
</table>

This register contains the 4GB aligned base address of gfx 4GB virtual address space within the host's 64-bit virtual address space of the instruction being fetched by the Primary DMA engine. This register contents are valid only when Batch Buffer is active.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td>** Reserved**</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
<tr>
<td>15:0</td>
<td></td>
<td><strong>Current DMA Address Upper DWORD</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: GraphicsAddress[47:32]</td>
</tr>
</tbody>
</table>

This field contains 4GB aligned base address of gfx 4GB virtual address space within the host's 64-bit virtual address space of the batch buffer that the "Primary" instruction parser DMA engine is currently accessing (fetching). Note that this address will typically lead the Head offset (as instructions must be fetched before execution).
**BCS Reported Timestamp Count**

<table>
<thead>
<tr>
<th>BCS_TIMESTAMP - BCS Reported Timestamp Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: BlitterCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000, 0x00000000</td>
</tr>
<tr>
<td>Access: RO. This register is not set by the context restore.</td>
</tr>
<tr>
<td>Size (in bits): 64</td>
</tr>
<tr>
<td>Address: 22358h</td>
</tr>
</tbody>
</table>

This register provides an elapsed real-time value that can be used as a timestamp. This register is not reset by a graphics reset. It will maintain its value unless a full chipset reset is performed. Note: This timestamp register reflects the value of the PCU TSC. The PCU TSC counts 10ns increments; this timestamp reflects bits 38:3 of the TSC (i.e. 80ns granularity, rolling over every 1.5 hours).

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>63:36</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: All</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
<tr>
<td>35:0</td>
<td></td>
<td>Timestamp Value</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: All</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: U36</td>
</tr>
</tbody>
</table>

This register toggles every 80 ns. The upper 28 bits are zero.
# BCS Reset Control Register

## BCS_RESET_CTRL - BCS Reset Control Register

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>BlitterCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>220D0h</td>
</tr>
</tbody>
</table>

This register is to be used to control soft reset.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td>Mask Bits</td>
</tr>
<tr>
<td></td>
<td>Format:</td>
<td>Mask[15:0]</td>
</tr>
<tr>
<td></td>
<td>Must be set to modify corresponding bit in Bits 15:0. (All implemented bits)</td>
<td></td>
</tr>
<tr>
<td>15:2</td>
<td>Reserved</td>
<td>MBZ</td>
</tr>
<tr>
<td>1</td>
<td>Ready for Reset</td>
<td>U1</td>
</tr>
<tr>
<td></td>
<td>Format:</td>
<td>U1</td>
</tr>
<tr>
<td></td>
<td>When set indicates blitter engine is ready for reset. This bit gets cleared on engine reset or when Soft Reset In progress is cleared.</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>Request Reset</td>
<td>U1</td>
</tr>
<tr>
<td></td>
<td>Format:</td>
<td>U1</td>
</tr>
<tr>
<td></td>
<td>When set, this bit indicates that SW wishes to reset the blitter engine. On seeing this bit set, Command Streamer will take appropriate action and set Ready For Reset status bit. This bit gets cleared on engine reset.</td>
<td></td>
</tr>
</tbody>
</table>
**BCS_RNCID - BCS Ring Buffer Next Context ID Register**

<table>
<thead>
<tr>
<th><strong>Register Space:</strong></th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Project:</strong></td>
<td>CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong></td>
<td>BlitterCS</td>
</tr>
<tr>
<td><strong>Default Value:</strong></td>
<td>0x00000000, 0x00000000</td>
</tr>
<tr>
<td><strong>Access:</strong></td>
<td>R/W</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong></td>
<td>64</td>
</tr>
<tr>
<td><strong>Address:</strong></td>
<td>22198h-2219Fh</td>
</tr>
</tbody>
</table>

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

**Programming Notes**

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

<table>
<thead>
<tr>
<th><strong>DWord</strong></th>
<th><strong>Bit</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>63:0</td>
<td><strong>Unnamed</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>See Context Descriptor for BCS</td>
</tr>
</tbody>
</table>
## BCS Semaphore Polling Interval on Wait

<table>
<thead>
<tr>
<th>BCS_SEMA_WAIT_POLL - BCS Semaphore Polling Interval on Wait</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space:</td>
</tr>
<tr>
<td>Project:</td>
</tr>
<tr>
<td>Source:</td>
</tr>
<tr>
<td>Default Value:</td>
</tr>
<tr>
<td>Access:</td>
</tr>
<tr>
<td>Size (in bits):</td>
</tr>
<tr>
<td>Address:</td>
</tr>
</tbody>
</table>

The SEMA_WAIT_POLL register contains Poll Interval field which specifies the minimum number of microseconds allowed for command streamer to wait before re-fetching the data from the address mentioned in the MI_SEMAPHORE_WAIT command on WAIT Mode set to POLL until the condition is satisfied while the context is not switched out. When value of 0 is written the poll interval will be equal to the memory latency of the read completion.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:21</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: All</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
<tr>
<td>20:0</td>
<td>20:0</td>
<td><strong>Poll Interval</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: All</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: U21</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Minimum number of micro-seconds allowed</td>
</tr>
</tbody>
</table>
# BCS SW Control

<table>
<thead>
<tr>
<th>BCS_SWCTRL - BCS SW Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: BlitterCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Access: r/w</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 22200h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td>Mask</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: WO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: Mask</td>
</tr>
<tr>
<td>0</td>
<td>15:4</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
<tr>
<td>0</td>
<td>3:2</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Tile Y Destination</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: U1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Programming this bit makes the HW treat all destination surfaces as Tile Y. This bit over-rides the setting of the destination format in the packet provided to the blitter command streamer. SW is required to flush the HW before changing the polarity of this bit. This bit is part of the context save/restore.</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>Tile Y Source</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: U1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Programming this bit makes the HW treat all source surfaces as Tile Y. This bit over-rides the setting of the source format in the packet provided to the blitter command streamer. SW is required to flush the HW before changing the polarity of this bit. This bit is part of the context save/restore.</td>
</tr>
</tbody>
</table>
# BCS Watchdog Counter Threshold

## BCS_CTR_THRSH - BCS Watchdog Counter Threshold

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>BlitterCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00150000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>2217Ch</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Counter logic Threshold</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00150000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: U32</td>
</tr>
</tbody>
</table>

This field specifies the threshold that the hardware checks against for the value of the blitter clock counter before generating an interrupt. The counter in hardware generates an interrupt when the threshold is reached, rolls over and starts counting again. The interrupt generated is the "Media Hang Notify" interrupt since this watchdog timer is intended primarily to remedy VLD hangs on the main pipeline.
Bitstream Output Bit Count for the last Syntax Element Report Register

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

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>VideoCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>RO</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Trusted Type:</td>
<td>1</td>
</tr>
<tr>
<td>Address:</td>
<td>128D4h</td>
</tr>
<tr>
<td>Name:</td>
<td>VDBOX1</td>
</tr>
<tr>
<td>Valid Projects:</td>
<td>CHV, BSW</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>MFC Bitstream Syntax Element Bit Count</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Total number of bits in the bitstream output before padding. This count is updated each time the internal counter is incremented.</td>
</tr>
</tbody>
</table>
Bitstream Output Byte Count Per Slice Report Register

**MFC_BITSTREAM_BYTECOUNT_SLICE - Bitstream Output Byte Count Per Slice Report Register**

- **Register Space:** MMIO: 0/2/0
- **Project:** CHV, BSW
- **Source:** VideoCS
- **Default Value:** 0x00000000
- **Access:** RO
- **Size (in bits):** 32
- **Trusted Type:** 1
- **Address:** 128D0h
- **Valid Projects:** CHV, BSW

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>MFC Bitstream Byte Count</strong>&lt;br&gt;Total number of bytes in the bitstream output from the encoder. This count is updated for every time the internal bitstream counter is incremented.</td>
</tr>
</tbody>
</table>
**Bitstream Output Minimal Size Padding Count Report Register**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>VideoCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>RO</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Trusted Type:</td>
<td>1</td>
</tr>
<tr>
<td>Address:</td>
<td>12814h</td>
</tr>
<tr>
<td>Name:</td>
<td>VDBOX1</td>
</tr>
<tr>
<td>Valid Projects:</td>
<td>CHV, BSW</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>MFC AVC MinSize Padding Count</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Total number of bytes in the bitstream output contributing to minimal size padding operation. This count is updated each time when the padding count is incremented.</td>
</tr>
</tbody>
</table>
## Blitter Mode Register

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space</strong></td>
<td>MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project</strong></td>
<td>CHV, BSW</td>
</tr>
<tr>
<td><strong>Source</strong></td>
<td>BlitterCS</td>
</tr>
<tr>
<td><strong>Default Value</strong></td>
<td>0x00000000</td>
</tr>
<tr>
<td><strong>Access</strong></td>
<td>R/W</td>
</tr>
<tr>
<td><strong>Size (in bits)</strong></td>
<td>32</td>
</tr>
<tr>
<td><strong>Trusted Type</strong></td>
<td>1</td>
</tr>
</tbody>
</table>

**Address:** 2229Ch

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>0</strong></td>
<td>31:16</td>
<td><strong>Mask Bits</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: Mask[15:0]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Must be set to modify corresponding bit in Bits 15:0. (All implemented bits)</td>
</tr>
<tr>
<td><strong>15</strong></td>
<td>15</td>
<td><strong>Execlist Enable</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Mask: MMIO#31</td>
</tr>
<tr>
<td></td>
<td></td>
<td>When set, software can utilize the execlist registers to load a context into hardware. When this bit is clear the Execlist mechanism cannot be used. The ring must be loaded via MMIO access.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Programming Notes</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>This bit is not intended to be changed dynamically. Changing the value of this bit while rendering is in progress will have UNDEFINED results. This bit should be changed only after a full reset and before submitting any commands to the device</td>
</tr>
<tr>
<td><strong>14</strong></td>
<td>14</td>
<td><strong>Interrupt Steering Bit</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: U1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>When set, Command Streamer sends interrupt messages to the SHIM of the ON CHIP Micro Controller through message channel. When reset, Command Streamer sends the interrupt messages to Display Engine as config writes on GAM interface.</td>
</tr>
<tr>
<td><strong>13:10</strong></td>
<td>13:10</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: All</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
</tbody>
</table>
## BLT_MODE - Blitter Mode Register

### 9 Per-Process GTT Enable

<table>
<thead>
<tr>
<th>Project:</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>Format:</td>
<td>Enable Per-Process GTT BS Mode Enable</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>PPGTT Disable</td>
<td>When clear, the Global GTT will be used to translate memory access from designated commands and for commands that select the PPGTT as their translation space.</td>
</tr>
<tr>
<td>1h</td>
<td>PPGTT Enable</td>
<td>When set, the PPGTT will be used to translate memory access from designated commands and for commands that select the PPGTT as their translation space.</td>
</tr>
</tbody>
</table>

**Programming Notes**

This bit is used for enabling PPGTT access in Ring Buffer mode of scheduling. Privilege field in context descriptor states the same in Execlist Mode of scheduling. This field should be set before programming PDP0/1/2/3 registers in order to set the PPGTT translation of memory access.

### 8 Reserved

<table>
<thead>
<tr>
<th>Project:</th>
<th>All</th>
</tr>
</thead>
</table>

### 7 64Bit Virtual Addressing Enable

<table>
<thead>
<tr>
<th>Project:</th>
<th>CHV, BSW</th>
</tr>
</thead>
<tbody>
<tr>
<td>Format:</td>
<td>Enable</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Per-Process GTT Enable</th>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0h</td>
<td>64Bit Virtual Addressing Disable</td>
<td>When clear indicates GFX operating in 32bit Virtual Addressing for PPGTT based memory access.</td>
</tr>
</tbody>
</table>

**Programming Notes**

This bit is only valid when PPGTT is enabled in ring buffer mode of scheduling. Context Descriptor has a similar bit to control 64bit virtual addressing in execlist mode of scheduling. Irrespective of this field set or clear virtual addresses translated through GGTT are always 32Bit. This field should be programmed before enabling PPGTT access. When this field is not set or for GGTT virtual addresses, Graphics Address [47:32] field of any commands or register exercised by SW should be programmed to 0x0.

64 Bit Virtual Addressing is not supported on CHV, BSW and must not be set.

### 6:5 Reserved

<table>
<thead>
<tr>
<th>Project:</th>
<th>CHV, BSW</th>
</tr>
</thead>
</table>

### 4 Reserved

<table>
<thead>
<tr>
<th>Project:</th>
<th>CHV, BSW</th>
</tr>
</thead>
</table>
### BLT_MODE - Blitter Mode Register

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3:1</td>
<td><strong>Reserved</strong>&lt;br&gt;Project: All&lt;br&gt;Format: MBZ</td>
</tr>
<tr>
<td>0</td>
<td><strong>Privilege Check Disable</strong>&lt;br&gt;Project: CHV, BSW&lt;br&gt;Format: Enable</td>
</tr>
</tbody>
</table>

This field when set, disables Privilege Violation checks on non-privileged batch buffers. When set, Privileged commands are allowed to be executed from non-privileged batch buffers.
## Blitter TLB Control Register

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:1</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0000000000000000000000000000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td><strong>Invalidate TLBs on the corresponding Engine</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

SW writes 1 to invalidate the TLBs for the associated engine and HW clears the bit when invalidation is complete. To ensure proper invalidation of the TLBs, SW has to ensure the corresponding engine’s HW pipeline is flushed and cleared from all its memory accesses. Otherwise HW cannot guarantee the proper invalidation for TLBs. This bit is self clear.
## BLT Context Element Descriptor (High Part)

<table>
<thead>
<tr>
<th>BLT_CTX_EDR_H - BLT Context Element Descriptor (High Part)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 04504h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>BLT Context Element Descriptor (High Part)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>


## BLT Context Element Descriptor (Low Part)

<table>
<thead>
<tr>
<th>BLT_CTX_EDR_L - BLT Context Element Descriptor (Low Part)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Registered Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000009</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 04500h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>BLT Context Element Descriptor (Low Part)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000009h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>
## BLT Context Element Descriptor (Low Part)

<table>
<thead>
<tr>
<th><strong>Register Space:</strong></th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Project:</strong></td>
<td>CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong></td>
<td>PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong></td>
<td>0x00000009</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong></td>
<td>32</td>
</tr>
<tr>
<td><strong>Address:</strong></td>
<td>04500h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>DWord</strong></th>
<th><strong>Bit</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>BLT Context Element Descriptor</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 00000009h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
</tbody>
</table>

**BLT Context Element Descriptor**
## BLT PDP0/PML4/PASID Descriptor (High Part)

**BLT_CTX_PDP0_H - BLT PDP0/PML4/PASID Descriptor (High Part)**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0450Ch</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>BLT PDP0/PML4/PASID Descriptor (High Part)</td>
<td>00000000h</td>
<td>R/W</td>
</tr>
</tbody>
</table>

**DWord: BLT_CTX_PDP0_H**

**Bit: 31:0**

**Description:** BLT PDP0/PML4/PASID Descriptor (High Part)

**Default Value:** 00000000h

**Access:** R/W
### BLT PDP0/PML4/PASID Descriptor (Low Part)

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>BLT PDP0/PML4/PASID Descriptor (Low Part)</td>
<td>00000000h</td>
<td>R/W</td>
</tr>
</tbody>
</table>

Register Space: MMIO: 0/2/0  
Project: CHV, BSW  
Source: PRM  
Default Value: 0x00000000  
Size (in bits): 32  
Address: 04508h
## BLT PDP1 Descriptor Register (High Part)

<table>
<thead>
<tr>
<th>BLT_CTX_PDP1_H - BLT PDP1 Descriptor Register (High Part)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space:</td>
</tr>
<tr>
<td>Project:</td>
</tr>
<tr>
<td>Source:</td>
</tr>
<tr>
<td>Default Value:</td>
</tr>
<tr>
<td>Size (in bits):</td>
</tr>
<tr>
<td>Address:</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>BLT PDP1 Descriptor Register (High Part)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>
**BLT PDP1 Descriptor Register (Low Part)**

<table>
<thead>
<tr>
<th>BLT_CTX_PDP1_L - BLT PDP1 Descriptor Register (Low Part)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 04510h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>DWORD</strong></th>
<th><strong>Bit</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>BLT PDP1 Descriptor Register (Low Part)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>R/W</td>
</tr>
</tbody>
</table>
BLT PDP2 Descriptor Register (High Part)

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>BLT PDP2 Descriptor Register (High Part)</strong></td>
</tr>
</tbody>
</table>

- **Default Value:** 00000000h
- **Access:** R/W

Register Space: MMIO: 0/2/0  
Project: CHV, BSW  
Source: PRM  
Default Value: 0x00000000  
Size (in bits): 32  
Address: 0451Ch
## BLT PDP2 Descriptor Register (Low Part)

### BLT_CTX_PDP2_L - BLT PDP2 Descriptor Register (Low Part)

<table>
<thead>
<tr>
<th>Register Space</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits)</td>
<td>32</td>
</tr>
<tr>
<td>Address</td>
<td>04518h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>BLT PDP2 Descriptor Register (Low Part)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>
BLT PDP3 Descriptor Register (High Part)

<table>
<thead>
<tr>
<th>BLT_CTX_PDP3_H - BLT PDP3 Descriptor Register (High Part)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 04524h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Word</strong></th>
<th><strong>Bit</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>BLT PDP3 Descriptor Register (High Part)</strong></td>
</tr>
</tbody>
</table>
### BLT_PDP3 Descriptor Register (Low Part)

<table>
<thead>
<tr>
<th>BLT_CTX_PDP3_L - BLT PDP3 Descriptor Register (Low Part)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 04520h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>BLT PDP3 Descriptor Register (Low Part)</td>
</tr>
</tbody>
</table>

- **Default Value:** 00000000h
- **Access:** R/W
### Boolean.Counter_B0

<table>
<thead>
<tr>
<th>register_space</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>project</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>source</td>
<td>BSpec</td>
</tr>
<tr>
<td>default_value</td>
<td>0x00000000</td>
</tr>
<tr>
<td>access</td>
<td>R/W</td>
</tr>
<tr>
<td>size (in bits)</td>
<td>32</td>
</tr>
<tr>
<td>address</td>
<td>02920h</td>
</tr>
<tr>
<td>valid_projects</td>
<td>[CHV, BSW]</td>
</tr>
</tbody>
</table>

This register enables the current live value of performance counter B0 to be read. Since what conditions cause B0 to increment are defined by the programming of CEC0-0/CEC0-1, please refer to the description of these registers and the values they hold to determine what event is reported via this register.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Considerations</strong></td>
</tr>
</tbody>
</table>

This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.
### Boolean_Counter_B1

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>BSpec</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>02924h</td>
</tr>
<tr>
<td>Valid Projects:</td>
<td>[CHV, BSW]</td>
</tr>
</tbody>
</table>

This register enables the current live value of performance counter B1 to be read. Since what conditions cause B1 to increment are defined by the programming of CEC1-0/CEC1-1, please refer to the description of these registers and the values they hold to determine what event is reported via this register.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Considerations</strong></td>
</tr>
</tbody>
</table>

This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.
### Boolean_Counter_B2

<table>
<thead>
<tr>
<th>OAPERF_B2 - Boolean_Counter_B2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: BSpec</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Access: R/W</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
</tbody>
</table>

Address: 02928h

Valid Projects: [CHV, BSW]

This register enables the current live value of performance counter B2 to be read. Since what conditions cause B2 to increment are defined by the programming of CEC0-0/CEC0-1, please refer to the description of these registers and the values they hold to determine what event is reported via this register.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Considerations</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format:</strong> U32</td>
</tr>
</tbody>
</table>

This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.
### Boolean Counter_B3

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>BSpec</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0292Ch</td>
</tr>
<tr>
<td>Valid Projects:</td>
<td>[CHV, BSW]</td>
</tr>
</tbody>
</table>

This register enables the current live value of performance counter B3 to be read. Since what conditions cause B3 to increment are defined by the programming of CEC0-0/CEC0-1, please refer to the description of these registers and the values they hold to determine what event is reported via this register.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Considerations</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: [U32]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no &quot;latch and hold&quot; mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.</td>
</tr>
</tbody>
</table>
Boolean.Counter_B4

**OAPERF_B4 - Boolean.Counter_B4**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>BSpec</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>02930h</td>
</tr>
<tr>
<td>Valid Projects:</td>
<td>[CHV, BSW]</td>
</tr>
</tbody>
</table>

This register enables the current live value of performance counter B4 to be read. Since what conditions cause B4 to increment are defined by the programming of CEC0-0/CEC0-1, please refer to the description of these registers and the values they hold to determine what event is reported via this register.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Considerations</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: U32</td>
</tr>
</tbody>
</table>

This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.
### Boolean Counter B5

**OAPERF_B5 - Boolean Counter B5**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>BSpec</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>02934h</td>
</tr>
<tr>
<td>Valid Projects:</td>
<td>[CHV, BSW]</td>
</tr>
</tbody>
</table>

This register enables the current live value of performance counter B5 to be read. Since what conditions cause B5 to increment are defined by the programming of CEC0-0/CEC0-1, please refer to the description of these registers and the values they hold to determine what event is reported via this register.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Considerations</strong></td>
</tr>
</tbody>
</table>

This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.
Boolean_Counter_B6

**OAPERF_B6 - Boolean_Counter_B6**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>BSpec</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>02938h</td>
</tr>
<tr>
<td>Valid Projects:</td>
<td>[CHV, BSW]</td>
</tr>
</tbody>
</table>

This register enables the current live value of performance counter B6 to be read. Since what conditions cause B6 to increment are defined by the programming of CEC0-0/CEC0-1, please refer to the description of these registers and the values they hold to determine what event is reported via this register.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Considerations</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format:</strong> U32</td>
</tr>
</tbody>
</table>

This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.
**Boolean-counter_B7**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>BSpec</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0293Ch</td>
</tr>
<tr>
<td>Valid Projects:</td>
<td>[CHV, BSW]</td>
</tr>
</tbody>
</table>

This register enables the current live value of performance counter B7 to be read. Since what conditions cause B7 to increment are defined by the programming of CEC0-0/CEC0-1, please refer to the description of these registers and the values they hold to determine what event is reported via this register.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Considerations</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: U32</td>
</tr>
</tbody>
</table>
|       |      | This 32-bit field returns bits 31:0 of the live performance counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.
### BTP_PRODUCE_COUNT - BTB Not Consumed By RCS

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>RenderCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Trusted Type:</td>
<td>1</td>
</tr>
<tr>
<td>Address:</td>
<td>02480h</td>
</tr>
</tbody>
</table>

This register keeps track of the outstanding BTP produced by RS which are not yet consumed by Render Command Streamer. This register is part of the render context save and restore.

#### Programming Notes

This register should not be programmed by SW.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>BTP Produce Count</strong></td>
</tr>
</tbody>
</table>

This register keeps track of the outstanding BTP produced by RS which are not yet consumed by Render Command Streamer. This register is part of the render context save and restore.
BTP Commands Parsed By RCS

<table>
<thead>
<tr>
<th>BTP_PARSE_COUNT - BTP Commands Parsed By RCS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: RenderCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Access: R/W</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Trusted Type: 1</td>
</tr>
<tr>
<td>Address: 02490h</td>
</tr>
</tbody>
</table>

This register keeps track of the BTP commands parsed by RCS prior to encountering the Draw Call in an RS enabled Batch Buffer which has BTP Pool Alloc Valid. BTP parse count should be less than equal to the BTP produce count for Command Streamer to make progress on a 3D_PRIMITIVE command. BTP Parse count is subtracted from the BTP Produce Count upon parsing 3D_PRIMITIVE command. This register is part of the render context save and restore. This register should not be programmed by SW.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>BTP Parse Count</strong></td>
</tr>
</tbody>
</table>

This register keeps track of the BTP commands parsed by RCS prior to encountering the Draw Call in an RS enabled Batch Buffer which has BTP Pool Alloc Valid. BTP parse count should be less than equal to the BTP produce count for Command Streamer to make progress on a 3D_PRIMITIVE command. BTP Parse count is subtracted from the BTP Produce Count upon parsing 3D_PRIMITIVE command.
# Cache Mode Register 0

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td>Mask</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: WO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: Mask[15:0]</td>
</tr>
</tbody>
</table>

A 1 in a bit in this field allows the modification of the corresponding bit in Bits 15:0.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td></td>
<td>Sampler L2 Disable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: r/w</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: Disable</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>[Default]</td>
<td>Sampler L2 Cache Enabled.</td>
</tr>
<tr>
<td>1h</td>
<td></td>
<td>Sampler L2 Cache Disabled. All accesses are treated as misses.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>14:12</td>
<td></td>
<td>MSAA Compression Plane Number Threshold for eLLC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Security: IP.eLLC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: r/w</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>threshold0 [Default]</td>
<td>Cache only planeID = 0 in eLLC.</td>
</tr>
<tr>
<td>1h</td>
<td>threshold1</td>
<td>Cache only planeID = 0, 1 in eLLC.</td>
</tr>
</tbody>
</table>

Description

This register is used to control the operation of the Render and Sampler L2 Caches. All reserved bits are implemented as read/write.

Before changing the value of this register, GFX pipeline must be idle i.e. full flush is required.

This Register is saved and restored as part of Context.

RegisterType = MMIO_SVL
### CACHE_MODE_0 - Cache Mode Register 0

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2h</td>
<td>threshold2</td>
<td>Cache only planeID = 0..2 in eLLC.</td>
</tr>
<tr>
<td>3h</td>
<td>threshold3</td>
<td>Cache only planeID = 0..3 in eLLC.</td>
</tr>
<tr>
<td>4h</td>
<td>threshold4</td>
<td>Cache only planeID = 0..4 in eLLC.</td>
</tr>
<tr>
<td>5h</td>
<td>threshold5</td>
<td>Cache only planeID = 0..5 in eLLC.</td>
</tr>
<tr>
<td>6h</td>
<td>threshold6</td>
<td>Cache only planeID = 0..6 in eLLC.</td>
</tr>
<tr>
<td>7h</td>
<td>threshold7</td>
<td>Cache only planeID = 0..7 in eLLC.</td>
</tr>
</tbody>
</table>

#### Programming Notes

This bit-field is programmed based on MSAA. When MSAA compression is enabled, these settings affect HW, else it is ignored. For 16X MSAA only lower 8 planes can be cached in eLLC.

### Sampler Set Remmapping for 3D Disable

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>Enable Set Remap [Default]</td>
<td>Set remapping for 3d enabled</td>
</tr>
<tr>
<td>1h</td>
<td>Disable Set Remap</td>
<td>Set remapping for 3d disabled</td>
</tr>
</tbody>
</table>

### Reserved

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>Enable [Default]</td>
<td>Allocation pipe is not stalled if there are pending expansions</td>
</tr>
<tr>
<td>1h</td>
<td>Disable</td>
<td>Allocation pipe is stalled if there are pending expansions</td>
</tr>
</tbody>
</table>

### Sampler L2 TLB Prefetch Enable

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>[Default]</td>
<td>TLB Prefetch Disabled</td>
</tr>
<tr>
<td>1h</td>
<td></td>
<td>TLB Prefetch Enabled</td>
</tr>
</tbody>
</table>

### Reserved

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### CACHE_MODE_0 - Cache Mode Register 0

#### 7:6 Sampler L2 Request Arbitration

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>00b</td>
<td>Round Robin</td>
<td></td>
</tr>
<tr>
<td>01b</td>
<td>Fetch are Highest Priority</td>
<td></td>
</tr>
<tr>
<td>10b</td>
<td>Constants are Highest Priority</td>
<td></td>
</tr>
<tr>
<td>11b</td>
<td>Reserved</td>
<td></td>
</tr>
</tbody>
</table>

| Project | CHV, BSW                     | Access: r/w                  | Format: U2                       |

#### 5 STC Eviction Policy

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

<table>
<thead>
<tr>
<th>Programming Notes</th>
<th>Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>If this bit is set to &quot;1&quot;, bit 4 of 0x7010h must also be set to &quot;1&quot;.</td>
<td>CHV, BSW</td>
</tr>
</tbody>
</table>

#### 4 RCC Eviction Policy

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

<table>
<thead>
<tr>
<th>Programming Notes</th>
<th>Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>If this bit is set to &quot;1&quot;, bit 7 of 0x7010h must also be set to &quot;1&quot;.</td>
<td>CHV, BSW</td>
</tr>
</tbody>
</table>

#### 3 Reserved
## CACHE_MODE_0 - Cache Mode Register 0

### Hierarchical Z RAW Stall Optimization Disable

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>Enable</td>
<td>Enables the hierarchical Z RAW Stall Optimization.</td>
</tr>
<tr>
<td>1h</td>
<td>Disable [Default]</td>
<td>Disables the hierarchical Z RAW Stall Optimization.</td>
</tr>
</tbody>
</table>

**Project:** CHV, BSW  
**Access:** r/w  
**Format:** U1

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

### Programming Notes

This bit must be set to 0 to enable the Hierarchical Z RAW stall optimization.

**Project:** CHV, BSW

### Disable clock gating in the pixel backend

| Access: | r/w  
| Format: | Disable

MCL related clock gating is disabled in the pixel backend. Before setting this bit to 1, the instruction/state caches must be invalidated. [DevGT:{WKA}]

### RW optimization (portal2) fix disable

| Project: | CHV, BSW  
| Access: | r/w

**Portal 2 fix:**

- a) In MSAA mode, RCC source clear cacheline will be allocated as RW, only if its a true read (ie blend enabled, etc) and will be allocated as WO otherwise.
- b) In any mode, RCC should not allocate a cacheline as RW, if its not a true read, evenif MSC sends RW = 11 to RCC. MSC sends an additional "True read" indicator to help RCC override the RW bit.

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 [Default]</td>
<td>Portal 2 fix enabled.</td>
<td></td>
</tr>
</tbody>
</table>
| 1 | Portal 2 fix disabled.  
  a) RCC will always allocate source clear as RW in MSAA mode.  
  b) RCC will always allocate on the basic of MSC RW indicator to RCC, in non source clear mode. |
# Cache Mode Register 1

**CACHE_MODE_1 - Cache Mode Register 1**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>RenderCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000180</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>07004h</td>
</tr>
<tr>
<td>Valid Projects:</td>
<td>CHV, BSW</td>
</tr>
</tbody>
</table>

## Description

RegisterType: MMIO_SVL

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

### DWord 0: Mask

- **Bit**: 31:16
- **Access**: WO
- **Mask**: MASK
- **Format**: Mask[15:0]

Must be set to modify corresponding data bit. Reads to this field returns zero.

### DWord 15: MSC Smart Bank Arbitration Disable

- **Project**: CHV, BSW
- **Access**: r/w
- **Format**: Disable

Setting this bit causes MSC Bank Arbitration to be disabled. Default value, i.e. resetting this bit, will Enable MSC Smart Bank Arbitration.

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>Enable [Default]</td>
</tr>
<tr>
<td>1h</td>
<td>Disable</td>
</tr>
</tbody>
</table>
### CACHE_MODE_1 - Cache Mode Register 1

#### MSC Resolve Optimization Disable
- **Project:** CHV, BSW
- **Access:** r/w
- **Format:** U1

Setting this bit causes MSC to mark cachelines dirty and appropriately update MSC during the classic clear resolve pass. Default value, i.e. resetting this bit, suppresses MSC buffer modification during the classic clear resolve pass.

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1h</td>
<td>Disable</td>
</tr>
<tr>
<td>0h</td>
<td>Enable [Default]</td>
</tr>
</tbody>
</table>

#### NP EARLY Z FAILS DISABLE
- **Project:** CHV, BSW
- **Access:** r/w

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>Disable [Default]</td>
<td>When NP PMA FIX ENABLE = 1, clearing this bit disables IZ to conservatively fail pixels.</td>
</tr>
<tr>
<td>1h</td>
<td>Enable</td>
<td>When NP PMA FIX ENABLE = 1, IZ does conservatively fail any NP pixels.</td>
</tr>
</tbody>
</table>

**Programming Notes**
- This bit must be set when NP PMA FIX ENABLE = 1
- This bit must not be set when NP PMA FIX ENABLE = 0

#### HIZ Eviction Policy
- **Project:** CHV, BSW
- **Access:** r/w
- **Format:** U1

If this bit is set, Hizunit will have LRA as replacement policy. The default value i.e. (when this bit is reset) indicates the non-LRA eviction policy. For performance reasons, this bit must be reset.

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>Non-LRA eviction Policy [Default]</td>
<td></td>
</tr>
<tr>
<td>1h</td>
<td>LRA eviction Policy</td>
<td></td>
</tr>
</tbody>
</table>

**Programming Notes**
- If this bit is set to "1", bit 3 of 0x7010h must also be set to "1"
## CACHE_MODE_1 - Cache Mode Register 1

### NP PMA FIX ENABLE

<table>
<thead>
<tr>
<th>Project</th>
<th>CHV, BSW</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access</td>
<td>r/w</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>Disable</td>
<td>Enables stalling PMA behavior for NP depth pixels in the early depth pipeline. (Legacy behavior)</td>
</tr>
<tr>
<td>1h</td>
<td>Enable</td>
<td>Enables non-stalling PMA behavior for NP depth pixels in the early depth pipeline.</td>
</tr>
</tbody>
</table>

**Programming Notes**

SW must set this bit in order to enable this fix when following expression is TRUE.

\[
\begin{align*}
3DSTATE_WM::ForceThreadDispatch &!= 1 \&\& !(3DSTATE_RASTER::ForceSampleCount != NUMRASTSAMPLES_0) \&\& (3DSTATE_DEPTH_BUFFER::SURFACE_TYPE != NULL) \&\& \\
(3DSTATE_DEPTH_BUFFER::HIZ Enable) \&\& !(3DSTATE_WM::EDSC_Mode == 2) \&\& \\
(3DSTATE_PS_EXTRA::PixelShaderValid) \&\& !(3DSTATE_WM_HZ_OP::DepthBufferResolve || 3DSTATE_WM_HZ_OP::Hierarchical Depth Buffer Resolve Enable || 3DSTATE_WM_HZ_OP::StencilBufferClear ) \&\& \\
(3DSTATE_WM_HZ_OP::DepthWriteEnable) \&\& ( \\
(3DSTATE_PS_EXTRA::PixelShaderKillsPixels || 3DSTATE_PS_EXTRA::oMask Present to RenderTarget || 3DSTATE_PS_BLEND::AlphaToCoverageEnable || \\
3DSTATE_PS_BLEND::AlphaTestEnable || 3DSTATE_WM_CHROMAKEY::ChromaKeyKillEnable ) \&\& (3DSTATE_WM::ForceKillPix != ForceOff) \&\& ( \\
(3DSTATE_WM_DEPTH_STENCIL::DepthWriteEnable) \&\& \\
3DSTATE_DEPTH_BUFFER::DEPTH_WRITE_ENABLE) \&\& (3DSTATE_WM_DEPTH_STENCIL::Stencil Buffer Write Enable) \&\& \\
(3DSTATE_STENCIL_BUFFER::STENCIL_BUFFER_ENABLE) ) ) || (3DSTATE_PS_EXTRA::Pixel Shader Computed Depth mode != PSCDEPTH_OFF) 
\end{align*}
\]

### Reserved

<table>
<thead>
<tr>
<th>Project</th>
<th>CHV, BSW</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access</td>
<td>r/w</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Project</th>
<th>CHV, BSW</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access</td>
<td>r/w</td>
</tr>
</tbody>
</table>
### CACHE_MODE_1 - Cache Mode Register 1

#### Sampler Cache Set XOR selection

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

**Programming Notes**

This field should be programmed as “00b” corresponding to NO XOR option when the 3D map performance fix in MT is enabled using the field “Sampler Set Remapping for 3D Disable” in CACHE_MODE_0 - Cache Mode Register 0.
## CACHE_MODE_1 - Cache Mode Register 1

### 4X4 RCPFE-STC Optimization Disable

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>[Default]</td>
<td>Enables two contiguous 4x2s to be collected as 4X4 access for STC interface. This allows for less bank collision and less RAM power on STC.</td>
</tr>
<tr>
<td>1h</td>
<td></td>
<td>Disables this optimization and therefore only one valid 4x2 is sent to STC on the 4X4 interface.</td>
</tr>
</tbody>
</table>

### MCS Cache Disable

For Programming restrictions please refer to the 3D Pipeline.

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

### Float Blend Optimization Enable

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>[Default]</td>
<td>Disables blend optimization for floating point RTs.</td>
</tr>
<tr>
<td>1h</td>
<td></td>
<td>Enables blend optimization for floating point RTs.</td>
</tr>
</tbody>
</table>

### Depth Read Hit Write-Only Optimization Disable

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>[Default]</td>
<td>Read Hit Write-only optimization is enabled in the Depth cache (RCZ).</td>
</tr>
<tr>
<td>1h</td>
<td></td>
<td>Read Hit Write-only optimization is disabled in the Depth cache (RCZ).</td>
</tr>
</tbody>
</table>
## CACHE_MODE_1 - Cache Mode Register 1

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>[Default]</td>
<td>RCZ will suppress the read request to memory if it was allocated as a expansion Cacheline. RCZ will always issue a read request to memory, even if it was previously allocated as expansion Cacheline.</td>
</tr>
<tr>
<td>1h</td>
<td></td>
<td>RCZ will always issue a read request to memory, even if it was previously allocated as expansion Cacheline. RCZ will suppress the read request to memory if it was allocated as a expansion Cacheline.</td>
</tr>
</tbody>
</table>

### Blend Optimization Fix Disable

<table>
<thead>
<tr>
<th>Project:</th>
<th>CHV, BSW</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access:</td>
<td>r/w</td>
</tr>
</tbody>
</table>

**Description**

This bit when reset, enables blend optimization fix. If this bit is set, it disables the blend optimization fix and may exhibit corruption on alpha components in the render target under some conditions.

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>[Default]</td>
</tr>
<tr>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

### Reserved

<table>
<thead>
<tr>
<th>Project:</th>
<th>CHV, BSW</th>
</tr>
</thead>
</table>
# CAPPOINT

## CAPPOINT - CAPPOINT

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>PCI: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x000000D0</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>00034h</td>
</tr>
</tbody>
</table>

This register points to a linked list of capabilities implemented by this device.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:8</td>
<td><strong>RESERVED</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td>7:0</td>
<td></td>
<td><strong>CAPABILITIES_POINTER</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: D0h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Once</td>
</tr>
</tbody>
</table>

The first item in the capabilities list is at address D0h (PMCS).
This register is programmed by BIOS during boot-up. Once written, this register becomes Read Only. This register can only be cleared by a Reset.
## CCK_CTL1 - CCK_CTL1

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00200000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>101100h</td>
</tr>
</tbody>
</table>

**BGF related programming.**
Note: This register should be the last one written between 10_1100, 10_1104, 10_1108. To be more specific, 10_1100[0] is the last one to be programmed between 10_1100h, 10_1104h and 10_1108h registers. The reason is that CZ clock does not stop on a frequency change and hence the fused values for the CZ clock domain must be the same and not intermittently transition to an invalid value, like all zeros. Changed for CHV, BSW.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:28</td>
<td><strong>czcount_30ns</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 0h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
</tbody>
</table>

**Description**

This register contains the number of cz clock cycles that it takes to make up 30ns.

### Frequency (MHz) | czcount_30ns
--- | ---
400 | 4'b1011
333.333 | 4'b1001
320 | 4'b0001
266.667 | 4'b0111
200 | 4'b0101
533.333 | 4'b1111
466.667 | 4'b1101
373.333 | 4'b1011
360 | 4'b1010
355.556 | 4'b1010
350 | 4'b1010

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Reserved</th>
</tr>
</thead>
<tbody>
<tr>
<td>27</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> RO</td>
</tr>
</tbody>
</table>

Reserved
CCK_CTL1 - CCK_CTL1

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>26:24</td>
<td>WellPUfreq</td>
</tr>
<tr>
<td></td>
<td>Default Value: 000b</td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

While Render or Media is being waking up from standby, clocks are allowed to be run for some time for x-contention removal. For CHV, BSW, there is a perceived di/dt risk by running this x-contention removal at full CU2X/CU frequency. This potential issue is amplified for CHV, BSW supports a 1:1 primary gear ratio and having a CU2X clock frequency of upto 2G.

- 0xx: x-contention removal time will use: CU2X/1 for 2x clock, CU/1 for 1X clock.
- 100: x-contention removal time will use a CU/4.
- 101: x-contention removal time will use a CU/8.
- 110: x-contention removal time will use a CU/16.
- 111: x-contention removal time will use a CU/32.

**Programming Notes**
Punit should never program this bitfield to 3'b000 and it should program this to 3'b110.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>23</td>
<td>CFG1Fuse0</td>
</tr>
<tr>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

bgsfsource_config1fuse0. By default, there is no frequency change. The set of BGF and SS registers are provided by fuses. When frequency change is required, this bit must be set to ‘1’ and Punit/driver must program the remainder of the clock crossing registers.

This bit must be last bit written between register 10_1100, 10_1104 and 10_1108.

The following registers are affected by this configuration bit: selecting between fuse values and configuration register:

- 0x10_1100[26:24]
- 0x10_1100[22:16]
- 0x10_1104[31:0]
- 0x10_1108[31:0]

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>22:20</td>
<td>CZ2CU_PTRSEP</td>
</tr>
<tr>
<td></td>
<td>Default Value: 010b</td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

bgf_cz2cu_ptrsep. Pointer separation for cz2cu BGF

- 000 = Pointer separation of 0
- 001 = Pointer separation of 1
- 010 = Pointer separation of 2 (Primary)
- 011 = Pointer separation of 3 (Secondary)
- 100 = Pointer separation of 4
- 101 = Pointer separation of 5
- 110 = Pointer separation of 6
- 111 = Pointer separation of 7
# CCK_CTL1 - CCK_CTL1

<table>
<thead>
<tr>
<th>Offset</th>
<th>Field</th>
<th>Default Value</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>19</td>
<td>Reserved</td>
<td>0b</td>
<td>RO</td>
<td>Reserved</td>
</tr>
<tr>
<td>18:16</td>
<td>CU2CZ_PTRSEP</td>
<td>0h</td>
<td>R/W</td>
<td>bgf_cu2cz_ptrsep. Pointer separation for cu2cz BGF</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>000 = Pointer separation of 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>001 = Pointer separation of 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>010 = Pointer separation of 2 (Primary)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>011 = Pointer separation of 3 (Secondary)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>100 = Pointer separation of 4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>101 = Pointer separation of 5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>110 = Pointer separation of 6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>111 = Pointer separation of 7</td>
</tr>
<tr>
<td>15:9</td>
<td>Reserved</td>
<td>00h</td>
<td>RO</td>
<td>Reserved</td>
</tr>
<tr>
<td>8</td>
<td>ClkSquashEn</td>
<td>0b</td>
<td>R/W</td>
<td>Clock Squash Enable. 0 = disabled. 1 = enabled. When this bit is enabled the</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>GFX clock squashing feature is enabled. Clock squashing is only allowed when</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>VNN is set to Vmin.</td>
</tr>
<tr>
<td></td>
<td>Programming Notes</td>
<td></td>
<td></td>
<td>This field must be zero at all times when in HPLLmode.</td>
</tr>
<tr>
<td>7:1</td>
<td>Reserved</td>
<td>0h</td>
<td>RO</td>
<td>Reserved</td>
</tr>
<tr>
<td>0</td>
<td>DblBufPulse</td>
<td>0h</td>
<td>WO</td>
<td>Reading from this register will always return a zero.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Punit will write a one into this bit in order to push the BGF parameters</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>into the double buffer during CPD/clock squash being enabled.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>A write to this register bit creates a one clock wide pulse</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(doublebufferpulse)to the CPunit.</td>
</tr>
</tbody>
</table>
# CCK_CTL2

## CCK_CTL2 - CCK_CTL2

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>101104h</td>
</tr>
</tbody>
</table>

CZ and CU ratios for BGFs and for common clock calculations. Changed for CHV, BSW.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td><strong>CZ2CU_CUratio</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>R/W</td>
</tr>
</tbody>
</table>

These 16 bits show the CU part of the BGF ratio between CZ and CU clock domain. CU_SquashAllow (=CZ2CU_CUratio) must always be less than or equal to CU_SquashWindow. When ClkSquashEn=0, CU_SquashAllow = CZ2CU_CUratio = CU_SquashWindow. 10_1108h will be dont care when ClkSquashEn=0. The effective/average squashed frequency will be determined by: (CU_SquashAllow/CU_SquashWindow) When ClkSquashEn=1 and in GPLLmode, 12 <= CZ2CU_CUratio(10_1104[31:16]) <= CU_SquashWindow(10_1108[31:16]) and multiples of 4. Effective CU frequency = CU ratio / (CU Squash Window) Actual CU frequency = UnsquashedFrequency * CU ratio / (CU Squash Window) If in GPLLmode, CU2x frequency = (400/10)*N, where N inside (2,50) (333/8)*N, where N inside (2,48) (320/8)*N, where N inside (2,50) (267/6)*N, where N inside (2,45) (200/5)*N, where N inside (2,50) The value programmed into this bitfield will be N*4. In HPLLmode: CZ2CU_CUratio = 720/(INTEGER*2). The HPLLvco (as determined by CCK_FUSE_REG0[1:0]) is divided by an INTEGER=(1,2,3,4,5,6,8,9,10) to create the CU2x frequency. CUclk=CU2xclk/2.
<table>
<thead>
<tr>
<th>15:0</th>
<th><strong>CZ2CU_CZratio</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Default Value: 0000h</td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

These 16 bits shows the CU part of the BGF ratio between CZ and CU clock domain.

\[ CZ2CU\_CZratio = \frac{UsyncPeriod}{CZperiod} \]

In GPLLmode, UsyncPeriod is expected to be \( (8^{*}\text{GPLLref\_in\_ns}) \). GPLLref\_in\_ns is expected to be

- \( CZperiod^{*5} \) for CZ200
- \( CZperiod^{*6} \) for CZ266
- \( CZperiod^{*8} \) for CZ320
- \( CZperiod^{*8} \) for CZ333
- \( CZperiod^{*10} \) for CZ400.

That makes

- \( CZ2CU\_CZratio = 8^{*}5 \) for CZ200 and GPLL mode.
- \( CZ2CU\_CZratio = 8^{*}6 \) for CZ266 and GPLL mode.
- \( CZ2CU\_CZratio = 8^{*}8 \) for CZ320 and GPLL mode.
- \( CZ2CU\_CZratio = 8^{*}8 \) for CZ333 and GPLL mode.
- \( CZ2CU\_CZratio = 8^{*}10 \) for CZ400 and GPLL mode.

The above only applies for GPLL mode and when UsyncPeriod = 8*GPLLref, CPunit will internally generate the following signal:

\[ CZ\_Com\_limit = CZ2CU\_CZratio - 1 \]

In HPLLmode: \( CZ2CU\_CZratio=720/(\text{INTEGER}\*1) \). The HPLLvco (as determined by

\[ \text{CCK\_FUSE\_REG0}[1:0] \]

is divided by an \text{INTEGER}=(1,2,3,4,5,6,8,9,10) to create the CZ frequency.
### CCK_CTL3 - CCK_CTL3

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00008000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>101108h</td>
</tr>
</tbody>
</table>

Graphics clock squash control. Changed for CHV, BSW.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td><strong>CU SQUASH WINDOW</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This register will always show the number of unsquashed clock cycles in a usync window. CU_SquashWindow = UsyncPeriod/Unsquashed_CUperiod. In GPLL mode, UsyncPeriod is expected to be (8<em>GPLLref_in_ns). CUperiod will be GPLLref_in_ns/Ratio. That makes, CU_SquashWindow = (8</em>GPLLref_in_ns)/(GPLLref_in_ns/Ratio) = 8<em>Ratio. Ratio is the multiplier from GPLLref to provide the CU clock frequency. Note: CU_SquashWindow = 8</em>Ratio only applies for GPLL mode when the usync period is 8*GPLL. In HPLL mode, clock squashing is not supported and 10_1100[8] must be zero at all times in HPLL mode.</td>
</tr>
<tr>
<td>15:0</td>
<td></td>
<td><strong>CU SQUASH INIT</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 8000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This register is used to move around the location of where the clocks are squashed. For initial design, this is always set to 8000h.</td>
</tr>
</tbody>
</table>
# CCK_CTL4 - CCK_CTL4

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>10110Ch</td>
</tr>
</tbody>
</table>

**CZclk 30ns TSV Generation**

DCN requires support for more CZclk frequencies

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:17</td>
<td><strong>SPARE</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td><strong>CZCLK 30ns Mechanism</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This bit defines whether we use the old counter mechanism for generating the 30ns pulse or use this new TSV scheme</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 - Use the old counter mechanism to determine the 30ns pulse (default)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 - Use the new TSV scheme with the Allow and Window values from this register</td>
</tr>
<tr>
<td>15:8</td>
<td></td>
<td><strong>CZclk HPLL VCOS IN 30ns</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>These bits determine the window in terms of the number of HPLL VCO clock cycles in 30ns</td>
</tr>
<tr>
<td>7:0</td>
<td></td>
<td><strong>CZclk HPLL DIV</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>These bits define the integer used to divide HPLLvco to get CZclk</td>
</tr>
</tbody>
</table>
# CLAIM_ER

## Register Summary

<table>
<thead>
<tr>
<th>Register Space</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits)</td>
<td>32</td>
</tr>
<tr>
<td>Address</td>
<td>182028h</td>
</tr>
</tbody>
</table>

### Claim Error Counter
- Counts the RM claim error (no RM claim) reported at CLAIM_ERROR field of EIR

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31</td>
<td><strong>CLAIM_ER_CLR</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: WO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Writing a '1' to this bit clears the CLAIM_ER_CTR, CTR_OVERFLOW and CLAIM_ERROR bit (EIR[0]). A read of this bit always returns 0</td>
</tr>
<tr>
<td>30:17</td>
<td><strong>RESERVED</strong></td>
<td>Default Value: 0000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td><strong>CTR_OVERFLOW</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>When set, indicates that the CLAIM_ER_CTR field of this register has overflowed (i.e., claim-error count has reached the value of 65536) The overflow is supplied by the RMbus master.</td>
</tr>
<tr>
<td>15:0</td>
<td><strong>CLAIM_ER_CTR</strong></td>
<td>Default Value: 0000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
</tbody>
</table>
Clipper Invocation Counter

<table>
<thead>
<tr>
<th>CL_INVOCATION_COUNT - Clipper Invocation Counter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: RenderCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000, 0x00000000</td>
</tr>
<tr>
<td>Access: R/W</td>
</tr>
<tr>
<td>Size (in bits): 64</td>
</tr>
<tr>
<td>Trusted Type: 1</td>
</tr>
<tr>
<td>Address: 02338h</td>
</tr>
<tr>
<td>Valid Projects:</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>63:32</td>
<td><strong>CL Invocation Count Report UDW</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Number of objects entering the clipper stage. Updated only when Statistics Enable is set in CLIP_STATE (see the Clipper Chapter in the 3D Volume.)</td>
</tr>
<tr>
<td>31:0</td>
<td><strong>CL Invocation Count Report LDW</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Number of objects entering the clipper stage. Updated only when Statistics Enable is set in CLIP_STATE (see the Clipper Chapter in the 3D Volume.)</td>
</tr>
</tbody>
</table>
Clipper Primitives Counter

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>RenderCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000, 0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>64</td>
</tr>
<tr>
<td>Trusted Type:</td>
<td>1</td>
</tr>
<tr>
<td>Address:</td>
<td>02340h</td>
</tr>
</tbody>
</table>

Valid Projects:

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>63:32</td>
<td><strong>Clipped Primitives Output Count UDW</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Total number of primitives output by the clipper stage. This count is updated for every primitive output by the clipper stage, as long as Statistics Enable is set in SF_STATE (see the Clipper and SF Chapters in the 3D Volume.)</td>
</tr>
<tr>
<td>31:0</td>
<td></td>
<td><strong>Clipped Primitives Output Count LDW</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Total number of primitives output by the clipper stage. This count is updated for every primitive output by the clipper stage, as long as Statistics Enable is set in SF_STATE (see the Clipper and SF Chapters in the 3D Volume.)</td>
</tr>
</tbody>
</table>
**CLKGATE Messaging Register for Clocking Unit**

**MSG_CLKGATE_GCP - CLKGATE Messaging Register for Clocking Unit**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>16</td>
</tr>
<tr>
<td>Address:</td>
<td>0802Ch</td>
</tr>
</tbody>
</table>

Message registers have bit-wise masking applied for writes. The register consists of 16 bits of data in [15:0], and 16 bits of corresponding masks in [31:16]. In order to set bit0, for example, the data would be 0x0001_0001. In order to clear bit0, for example, the data would be 0x0001_0000. Note that mask bit is the data bit offset + 16. Message registers are protected from non-GT writes via the Message Channel.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>15:7</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>Gate cmclk Acknowledgement (VCS1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Gate cmclk Acknowledgement (VCS1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1'b0 : Clocks are ungated &lt;default&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1'b1 : Clocks are gated</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>Gate cwclk Acknowledgement (WIN)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Gate cwclk Acknowledgement (WIN)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1'b0 : Clocks are ungated &lt;default&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1'b1 : Clocks are gated</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>Reserved</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>Gate cfclk Acknowledgement (CS)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Gate cfclk Acknowledgement (CS)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1'b0 : Clocks are ungated &lt;default&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1'b1 : Clocks are gated</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>Gate cvclk Acknowledgement (VECS)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Gate cvclk Acknowledgement (VECS)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1'b0 : Clocks are ungated &lt;default&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1'b1 : Clocks are gated</td>
</tr>
</tbody>
</table>
**MSG_CLKGATE_GCP - CLKGATE Messaging Register for Clocking Unit**

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
<th>Access</th>
<th>Details</th>
</tr>
</thead>
</table>
| 1     | Gate cmclk Acknowledgement (VCS0) | R/W    | 1'b0 : Clocks are ungated <default>  
1'b1 : Clocks are gated |
| 0     | Gate crclk Acknowledgement (CS)   | R/W    | 1'b0 : Clocks are ungated <default>  
1'b1 : Clocks are gated |
# CLOCK_GATE_DIS1

## CLOCK_GATE_DIS1 - CLOCK_GATE_DIS1

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>182060h</td>
</tr>
</tbody>
</table>

**Gunit Clock Gating Disable**

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31</td>
<td><strong>GIOSF_DIS</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 = Enable clock gating</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 = Disable clock gating</td>
</tr>
<tr>
<td>0</td>
<td>30</td>
<td><strong>GPFI_DIS</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 = Enable clock gating</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 = Disable clock gating</td>
</tr>
<tr>
<td>0</td>
<td>29</td>
<td><strong>GRMW_DIS</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 = Enable clock gating</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 = Disable clock gating</td>
</tr>
<tr>
<td>0</td>
<td>28</td>
<td><strong>GSA_OUTBOUND_DIS</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 = Enable clock gating</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 = Disable clock gating</td>
</tr>
<tr>
<td>0</td>
<td>27</td>
<td><strong>GSA_RTN_DIS</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 = Enable clock gating</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 = Disable clock gating</td>
</tr>
<tr>
<td>Register</td>
<td>Default Value</td>
<td>Access</td>
</tr>
<tr>
<td>--------------------</td>
<td>---------------</td>
<td>--------</td>
</tr>
<tr>
<td>GUP_DIS</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td>GWAKE_DIS</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td>GCR_DIS</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td>GBC_DIS</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td>GINT_DIS</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td>GRMBUS_DIS</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td>GCCBCZ_KEYCMP_DIS</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td>Field</td>
<td>Default Value</td>
<td>Access</td>
</tr>
<tr>
<td>-----------------</td>
<td>---------------</td>
<td>--------</td>
</tr>
<tr>
<td>19:16 SPARE1</td>
<td>0h</td>
<td>R/W</td>
</tr>
<tr>
<td>15 Reserved</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14 Reserved</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13 Reserved</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12 GSECDISP_DIS</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11 GREGDISP_DIS</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10 GCTLQ_DIS</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9 GMMIO_DIS</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8 GCFG_DIS</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7:4 SPARE2</td>
<td>0h</td>
<td>R/W</td>
</tr>
</tbody>
</table>
### CLOCK_GATE_DIS1 - CLOCK_GATE_DIS1

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3:0</td>
<td><strong>CZ_IDLE_TIMER</strong></td>
</tr>
<tr>
<td>Default</td>
<td><strong>0h</strong></td>
</tr>
<tr>
<td>Access</td>
<td><strong>R/W</strong></td>
</tr>
</tbody>
</table>

Gunit CZ domain idle timer count value.
Individual Gunit CZ fubs use this as a counter pre-load value.
0 = Idle timer disabled
1 - 15 = After a unit indicates idle, wait this many clocks before gating.
## CLOCK_GATE_DIS2

**Register Space:** MMIO: 0/2/0  
**Project:** CHV, BSW  
**Source:** PRM  
**Default Value:** 0x00000000  
**Size (in bits):** 32

**Address:** 182064h

Gunit Clock Gating Disable register2.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:9</td>
<td><strong>CLKGATE_SPARE</strong></td>
<td>000000h</td>
<td>R/W</td>
<td>Spare to be assigned to clock gate disables.</td>
</tr>
</tbody>
</table>
| 8     | 7   | **GPMH_DIS**      | 0b            | R/W    | New for CHV, BSW  
|       |      |                   | 0 = Enable clock gating  
|       |      |                   | 1 = Disable clock gating |
| 7     | 6   | **ATOMICS_DIS**   | 0b            | R/W    | New for CHV, BSW  
|       |      |                   | 0 = Enable clock gating  
|       |      |                   | 1 = Disable clock gating |
| 6     | 5   | **IOSFSB_P_DIS**  | 0b            | R/W    | New for CHV, BSW  
|       |      |                   | 0 = Enable clock gating  
|       |      |                   | 1 = Disable clock gating |
| 5     | 4   | **IOSFSB_DIS**    | 0b            | R/W    | New for CHV, BSW  
|       |      |                   | 0 = Enable clock gating  
<p>|       |      |                   | 1 = Disable clock gating |</p>
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>GARB_DIS</td>
<td>0b</td>
<td>R/W</td>
<td>0 = Enable clock gating</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1 = Disable clock gating</td>
</tr>
<tr>
<td>3</td>
<td>GMAP_DIS</td>
<td>0b</td>
<td>R/W</td>
<td>0 = Enable clock gating</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1 = Disable clock gating</td>
</tr>
<tr>
<td>2</td>
<td>Reserved</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>GRDDATBUF_DIS</td>
<td>0b</td>
<td>R/W</td>
<td>0 = Enable clock gating</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1 = Disable clock gating</td>
</tr>
<tr>
<td>0</td>
<td>GWRDATQ_DIS</td>
<td>0b</td>
<td>R/W</td>
<td>0 = Enable clock gating</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1 = Disable clock gating</td>
</tr>
</tbody>
</table>
## Clock gate and clear EU metrics

### EUMETRICSCLEAR - Clock gate and clear EU metrics

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>138198h</td>
</tr>
</tbody>
</table>

#### EU Metrics Clear

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:1</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

- **Default Value:** 00000000h
- **Access:** RO

**Clear EU Metrics**

- **Default Value:** 0b
- **Access:** R/W

For Cdyn optimization, when this bit is set to one, all of the 13 counters are clock gated. Then, after setting it to '0(zero), On the falling edge of EUMetricsClr, all of the 13 counters are reset to zero and counting resumes.

<table>
<thead>
<tr>
<th>Event Name</th>
<th>Units</th>
<th>Maximum per cycle (average)</th>
<th>Metric</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>event0: ss0_gpm_sampler_active_igpa</td>
<td>1 = 1 sampler busy in this sub-slice.</td>
<td>1 sampler busy in this sub-slice.</td>
<td>Sampler busy per sub-slice</td>
<td>Measures sampler utilization (i.e, sampler not idle)</td>
</tr>
<tr>
<td>event1: ss1_gpm_sampler_active_igpa</td>
<td>1 = 1 sampler busy in this sub-slice.</td>
<td>1 sampler busy in this sub-slice.</td>
<td>Sampler busy per sub-slice</td>
<td>Measures sampler utilization (i.e, sampler not idle)</td>
</tr>
<tr>
<td>event2: tdl0_gpm_eu_stall</td>
<td>1 = 32 EUs stalled on this sub-slice.</td>
<td>8 EUs stalled on this sub-slice.</td>
<td>EU stall per sub-slice</td>
<td>Measures EU stalls</td>
</tr>
<tr>
<td>event3: tdl1_gpm_eu_stall</td>
<td>1 = 32 EUs stalled on this sub-slice.</td>
<td>8 EUs stalled on this sub-slice.</td>
<td>EU stall per sub-slice</td>
<td>Measures EU stalls</td>
</tr>
</tbody>
</table>
### EUMETRICS CLEAR - Clock gate and clear EU metrics

<table>
<thead>
<tr>
<th>Event ID</th>
<th>Description</th>
<th>Value</th>
<th>IPC per sub-slice</th>
<th>EU Utilization</th>
</tr>
</thead>
<tbody>
<tr>
<td>event4: tdl0_gpm_ipc_count</td>
<td>1 = 32 instructions executed on this sub-slice.</td>
<td>IPC per sub-slice</td>
<td>Measures EU utilization: 0, 1, or 2 counts per clock per EU</td>
<td></td>
</tr>
<tr>
<td>event5: tdl1_gpm_ipc_count</td>
<td>1 = 32 instructions executed on this sub-slice.</td>
<td>IPC per sub-slice</td>
<td>Measures EU utilization: 0, 1, or 2 counts per clock per EU</td>
<td></td>
</tr>
<tr>
<td>event6: tdl0_gpm_threads_allocated</td>
<td>1 = 64 threads allocated in this sub-slice.</td>
<td>Thread allocated per sub-slice</td>
<td>Measures the threads allocated</td>
<td></td>
</tr>
<tr>
<td>event7: tdl1_gpm_threads_allocated</td>
<td>1 = 64 threads allocated in this sub-slice.</td>
<td>Thread allocated per sub-slice</td>
<td>Measures the threads allocated</td>
<td></td>
</tr>
<tr>
<td>event8: tdl0_gpm_eu_stall_sampler</td>
<td>1 = 64 EUs stalled by sampler in this sub-slice.</td>
<td>Placeholder for EUstallTime by sampler per sub-slice</td>
<td>Measures how much of eu stalls are due to sampler</td>
<td></td>
</tr>
<tr>
<td>event9: tdl1_gpm_eu_stall_sampler</td>
<td>1 = 64 EUs stalled by sampler in this sub-slice.</td>
<td>Placeholder for EUstallTime by sampler per sub-slice</td>
<td>Measures how much of eu stalls are due to sampler</td>
<td></td>
</tr>
<tr>
<td>event10: tdl0_gpm_eu_not_idle</td>
<td>1 = 32 EUs not idle in this sub-slice.</td>
<td>Clocks any EU is not idle per sub-slice</td>
<td>Measures EU utilization (i.e, EU not idle)</td>
<td></td>
</tr>
<tr>
<td>event11: tdl1_gpm_eu_not_idle</td>
<td>1 = 32 EUs not idle in this sub-slice.</td>
<td>Clocks any EU is not idle per sub-slice</td>
<td>Measures EU utilization (i.e, EU not idle)</td>
<td></td>
</tr>
<tr>
<td>event12: constant_one</td>
<td>1</td>
<td></td>
<td>one clock cycle toggles.</td>
<td></td>
</tr>
</tbody>
</table>
## Clock Gating Messages

### CGMSG - Clock Gating Messages

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>08104h</td>
</tr>
</tbody>
</table>

### Clock Gating Messages Register

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td><strong>Message Mask</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>Access:</code> RO</td>
</tr>
<tr>
<td></td>
<td>15:7</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>Access:</code> RO</td>
</tr>
<tr>
<td>6</td>
<td>31:16</td>
<td><strong>Media 1 Clock gating control message</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>Access:</code> R/W</td>
</tr>
<tr>
<td>5</td>
<td>31:16</td>
<td><strong>WIDI Clock Gating control Message</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>Access:</code> R/W</td>
</tr>
<tr>
<td>4</td>
<td>31:16</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td>3</td>
<td>31:16</td>
<td><strong>Fix Function Clock gating Control Message</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>Access:</code> R/W</td>
</tr>
<tr>
<td>2</td>
<td>31:16</td>
<td><strong>VEbox Clock gating Control message</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>Access:</code> R/W</td>
</tr>
</tbody>
</table>

**Message Mask**  
In order to write to bits 15:0, the corresponding message mask bits must be written. For example, for bit 14 to be set, bit 30 needs to be 1: 40004000

**Media 1 Clock gating control message**  
Gate Media 1 (2nd Vbox) Clock Message:  
'0': Media 1 Clock Un-gate Request (un-gates the cmclk clock in the 2n Media block)  
'1': Media 1 Clock Gate Request (gates the cmclk clock in the 2nd Media block)

**WIDI Clock Gating control Message**  
Gate WIDI Clock Message:  
'0': WIDI Clock Un-gate Request (un-gates the cwclk clock)  
'1': WIDI Clock Gate Request (gates the cwclk clock)

**Fix Function Clock gating Control Message**  
Gate Fix Clock Message:  
'0': Fix Clock Un-gate Request (un-gates the cfclk/cf2xclk clock)  
'1': Fix Clock Gate Request (gates the cfclk/cf2xclk clock)

**VEbox Clock gating Control message**  
Gate VE-box Clock Message:  
'0': VEbox Clock Un-gate Request (un-gates the cvclk clock)  
'1': VEbox Clock Gate Request (gates the cvclk clock)
## CGMSG - Clock Gating Messages

<table>
<thead>
<tr>
<th></th>
<th>Media 0 Clock Gating Control Message</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td>Gate Media Clock Message:</td>
</tr>
<tr>
<td>1</td>
<td>'0' : Media 0 Clock Un-gate Request (un-gates the cmclk clock)</td>
</tr>
<tr>
<td></td>
<td>'1' : Media 0 Clock Gate Request (gates the cmclk clock)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Row Clock Gating Control Message</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td>Gate Row Clocks Message:</td>
</tr>
<tr>
<td>0</td>
<td>'0' : Row Clock Un-gate Request (un-gates the crclk and cr2xclk clocks)</td>
</tr>
<tr>
<td></td>
<td>'1' : Row Clock Gate Request (gates the crclk and cr2xclk clocks)</td>
</tr>
</tbody>
</table>
## Color/Depth Write FIFO Watermarks

<table>
<thead>
<tr>
<th>Description</th>
<th>DWord</th>
<th>Bit</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CZWMRK - Color/Depth Write FIFO Watermarks</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Register Space:</td>
<td>MMIO:</td>
<td>0/2/0</td>
<td></td>
<td>MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project:</td>
<td>CHV,</td>
<td>BSW</td>
<td></td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>Render</td>
<td>CS</td>
<td></td>
<td>RenderCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00</td>
<td>00000</td>
<td>000000</td>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
<td></td>
<td></td>
<td>Size: 32</td>
</tr>
<tr>
<td>Trusted Type:</td>
<td>1</td>
<td></td>
<td></td>
<td>Trusted Type: 1</td>
</tr>
<tr>
<td>Address:</td>
<td>04060h</td>
<td></td>
<td></td>
<td>Address: 04060h</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>Reserved</strong></td>
<td>MBZ</td>
<td>MBZ</td>
</tr>
<tr>
<td></td>
<td>23:18</td>
<td><strong>Color Wr Burst Size</strong></td>
<td>MBZ</td>
<td>This is the maximum size of the requests burst, from the last High Watermark trip, before reevaluating the High Watermark again.</td>
</tr>
<tr>
<td></td>
<td>17:16</td>
<td><strong>Reserved</strong></td>
<td>MBZ</td>
<td>MBZ</td>
</tr>
<tr>
<td></td>
<td>15:12</td>
<td><strong>Color Wr FIFO High Watermark</strong></td>
<td>MBZ</td>
<td>This is the number of accumulated Color writes that will trigger a Burst of Z Writes.</td>
</tr>
<tr>
<td></td>
<td>11:6</td>
<td><strong>Z Wr Burst Size</strong></td>
<td>MBZ</td>
<td>This is the maximum size of the requests burst, from the last High Watermark trip, before reevaluating the High Watermark again.</td>
</tr>
<tr>
<td></td>
<td>5:4</td>
<td><strong>Reserved</strong></td>
<td>MBZ</td>
<td>MBZ</td>
</tr>
<tr>
<td></td>
<td>3:0</td>
<td><strong>Z Wr FIFO High Watermark</strong></td>
<td>MBZ</td>
<td>This is the number of accumulated Depth writes that will trigger a Burst of Z Writes.</td>
</tr>
</tbody>
</table>
## Configuration Register0 for RPMunit

<table>
<thead>
<tr>
<th>Configuration Register0 for RPMunit</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 00D00h</td>
</tr>
</tbody>
</table>

Lock bit LOCK applies to all RW/L fields in this register. Lock is overridden during context restore.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31</td>
<td><strong>Lock for RW/L Fields in this Register</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 = Bits of CONFIG0 register are R/W.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 = All bits of CONFIG0 register are RO (including this lock bit).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Once written to 1, the lock is set and cannot be cleared (i.e., writing a 0 will not clear the lock).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Lock is reset on a restore after context is captured.</td>
</tr>
<tr>
<td>30:0</td>
<td></td>
<td><strong>Placeholder Bits</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Placeholder bits for implementation or ECO loops.</td>
</tr>
</tbody>
</table>
## Configuration Register1 for RPMunit

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31</td>
<td><strong>Lock for RW/L Fields in this Register</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 = Bits of CONFIG0 register are R/W.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 = All bits of CONFIG0 register are RO (including this lock bit).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Once written to 1, the lock is set and cannot be cleared (i.e., writing a 0 will not clear the lock).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Lock is reset on a restore after context is captured.</td>
</tr>
<tr>
<td>30:0</td>
<td></td>
<td><strong>Placeholder Bits</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Placeholder bits for implementation or ECO loops.</td>
</tr>
</tbody>
</table>
# Configuration Register for RCPunit

## Register Details

- **Register**: RCPCONFIG - Configuration Register for RCPunit
- **Register Space**: MMIO: 0/2/0
- **Project**: CHV, BSW
- **Default Value**: 0x0000000F
- **Size (in bits)**: 32
- **Address**: 00D08h

## Register Description

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:5</td>
<td><strong>Placeholder Bits</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td><strong>RPMunit Clock Gating Disable in Uncore Well</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always) (DEFAULT).</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td><strong>MGRSunit Clock Gating Disable in Uncore Well</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always) (DEFAULT).</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td><strong>MDRBunit Clock Gating Disable in Uncore Well</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always) (DEFAULT).</td>
</tr>
<tr>
<td>RCPCONFIG - Configuration Register for RCPunit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>0</strong></td>
<td><strong>MCRunit Clock Gating Disable in Uncore Well</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default Value: 1b</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
<td></td>
</tr>
</tbody>
</table>

Clock Gating Disable Control:

'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality).

'1': Clock Gating Disabled. (i.e., clocks are toggling, always) (DEFAULT).
## Context Load Protocol Register BLT

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td><strong>Mask Bits</strong></td>
<td>0000h</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td></td>
<td><strong>Context Load Protocol Register - BCS 15</strong></td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td><strong>Context Load Protocol Register - BCS 14</strong></td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
<td><strong>Context Load Protocol Register - BCS 13</strong></td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td><strong>Context Load Protocol Register - BCS 12</strong></td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td><strong>Context Load Protocol Register - BCS 11</strong></td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLT_CTX_LD_PRTCL - Context Load Protocol Register BLT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------------------------------------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>10</strong> Context Load Protocol Register - BCS 10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default Value:</td>
<td>0b</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>For Future Use.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>This bit is self clear.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>9</strong> Context Load Protocol Register - BCS 9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default Value:</td>
<td>0b</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>For Future Use.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>This bit is self clear.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>8</strong> Context Load Protocol Register - BCS 8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default Value:</td>
<td>0b</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>For Future Use.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>This bit is self clear.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>7</strong> Context Load Protocol Register - BCS 7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default Value:</td>
<td>0b</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>For Future Use.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>This bit is self clear.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>6</strong> Context Load Protocol Register - BCS 6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default Value:</td>
<td>0b</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>For Future Use.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>This bit is self clear.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>5</strong> Context Load Protocol Register - BCS 5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default Value:</td>
<td>0b</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>For Future Use.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>This bit is self clear.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>4</strong> Context Load Protocol Register - BCS 4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default Value:</td>
<td>0b</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>For Future Use.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>This bit is self clear.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>3</strong> Context Load Protocol Register - BCS 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default Value:</td>
<td>0b</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## BLT_CTX_LD_PRTCL - Context Load Protocol Register BLT

<table>
<thead>
<tr>
<th>Bit 3 = Context Complete. Only valid with Bit 2 asserted and indicates whether the context switching out as complete vs. incomplete. This bit is self clear.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>2</strong> Context Load Protocol Register - BCS 2</td>
</tr>
<tr>
<td>Default Value:</td>
</tr>
<tr>
<td>Access:</td>
</tr>
<tr>
<td>Context Load Protocol Register (Written by BCS) Bit 2 = Request from BCS to GAM for context save readiness. GAM will acknowledge appropriated steps taken. This bit is self clear.</td>
</tr>
<tr>
<td><strong>1</strong> Context Load Protocol Register - BCS 1</td>
</tr>
<tr>
<td>Default Value:</td>
</tr>
<tr>
<td>Access:</td>
</tr>
<tr>
<td>Context Load Protocol Register (Written by BCS) Bit 1 = Context Launched. This bit is self clear.</td>
</tr>
<tr>
<td><strong>0</strong> Context Load Protocol Register - BCS 0</td>
</tr>
<tr>
<td>Default Value:</td>
</tr>
<tr>
<td>Access:</td>
</tr>
<tr>
<td>Context Load Protocol Register (Written by BCS) Bit 0 = Context Available. This bit is self clear.</td>
</tr>
</tbody>
</table>
## GFX_CTX_LD_PRTCL - Context Load Protocol Register CS

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td>Mask Bits</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>Context Load Protocol Register - CS 15</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For Future Use. This bit is self clear.</td>
</tr>
<tr>
<td>14</td>
<td></td>
<td>Context Load Protocol Register - CS 14</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For Future Use. This bit is self clear.</td>
</tr>
<tr>
<td>13</td>
<td></td>
<td>Context Load Protocol Register - CS 13</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For Future Use. This bit is self clear.</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>Context Load Protocol Register - CS 12</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For Future Use. This bit is self clear.</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>Context Load Protocol Register - CS 11</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For Future Use. This bit is self clear.</td>
</tr>
</tbody>
</table>
### GFX_CTX_LD_PRTCL - Context Load Protocol Register CS

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>Context Load Protocol Register - CS 10</td>
<td>0b</td>
<td>R/W</td>
<td>For Future Use. This bit is self clear.</td>
</tr>
<tr>
<td>9</td>
<td>Context Load Protocol Register - CS 9</td>
<td>0b</td>
<td>R/W</td>
<td>For Future Use. This bit is self clear.</td>
</tr>
<tr>
<td>8</td>
<td>Context Load Protocol Register - CS 8</td>
<td>0b</td>
<td>R/W</td>
<td>For Future Use. This bit is self clear.</td>
</tr>
<tr>
<td>7</td>
<td>Context Load Protocol Register - CS 7</td>
<td>0b</td>
<td>R/W</td>
<td>For Future Use. This bit is self clear.</td>
</tr>
<tr>
<td>6</td>
<td>Context Load Protocol Register - CS 6</td>
<td>0b</td>
<td>R/W</td>
<td>For Future Use. This bit is self clear.</td>
</tr>
<tr>
<td>5</td>
<td>Context Load Protocol Register - CS 5</td>
<td>0b</td>
<td>R/W</td>
<td>For Future Use. This bit is self clear.</td>
</tr>
<tr>
<td>4</td>
<td>Context Load Protocol Register - CS 4</td>
<td>0b</td>
<td>R/W</td>
<td>For Future Use. This bit is self clear.</td>
</tr>
<tr>
<td>3</td>
<td>Context Load Protocol Register - CS 3</td>
<td>0b</td>
<td>CHV, BSW</td>
<td>R/W</td>
</tr>
</tbody>
</table>

**Access:**

- **R/W:** Read/Write.
<table>
<thead>
<tr>
<th>Bit 3 = Context Complete. Only valid with Bit 2 asserted and indicates whether the context switching out as complete vs. incomplete. This bit is self clear.</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 Context Load Protocol Register - CS 2</td>
</tr>
<tr>
<td>Default Value:</td>
</tr>
<tr>
<td>Access:</td>
</tr>
<tr>
<td>Context Load Protocol Register (Written by CS) Bit 2 = Request from CS to GAM for context save readiness. GAM will acknowledge appropriated steps taken. This bit is self clear.</td>
</tr>
<tr>
<td>1 Context Load Protocol Register - CS 1</td>
</tr>
<tr>
<td>Default Value:</td>
</tr>
<tr>
<td>Access:</td>
</tr>
<tr>
<td>Context Load Protocol Register (Written by CS) Bit 1 = Context Launched. This bit is self clear.</td>
</tr>
<tr>
<td>0 Context Load Protocol Register - CS 0</td>
</tr>
<tr>
<td>Default Value:</td>
</tr>
<tr>
<td>Access:</td>
</tr>
<tr>
<td>Context Load Protocol Register (Written by CS) Bit 0 = Context Available. This bit is self clear.</td>
</tr>
</tbody>
</table>
# Context Load Protocol Register VCS0

<table>
<thead>
<tr>
<th>MFX0_CTX_LD_PRTCL - Context Load Protocol Register VCS0</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000 CHV, BSW</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 04008h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td><strong>Mask Bits</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 0000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td><strong>Context Load Protocol Register - VCS0 15</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For Future Use.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This bit is self clear.</td>
</tr>
<tr>
<td>14</td>
<td></td>
<td><strong>Context Load Protocol Register - VCS0 14</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For Future Use.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This bit is self clear.</td>
</tr>
<tr>
<td>13</td>
<td></td>
<td><strong>Context Load Protocol Register - VCS0 13</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For Future Use.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This bit is self clear.</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td><strong>Context Load Protocol Register - VCS0 12</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For Future Use.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This bit is self clear.</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td><strong>Context Load Protocol Register - VCS0 11</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For Future Use.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This bit is self clear.</td>
</tr>
<tr>
<td>Register</td>
<td>Description</td>
<td>Default Value</td>
</tr>
<tr>
<td>----------</td>
<td>------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>VCS0 10</td>
<td>Context Load Protocol Register</td>
<td>0b</td>
</tr>
<tr>
<td>VCS0 9</td>
<td>Context Load Protocol Register</td>
<td>0b</td>
</tr>
<tr>
<td>VCS0 8</td>
<td>Context Load Protocol Register</td>
<td>0b</td>
</tr>
<tr>
<td>VCS0 7</td>
<td>Context Load Protocol Register</td>
<td>0b</td>
</tr>
<tr>
<td>VCS0 6</td>
<td>Context Load Protocol Register</td>
<td>0b</td>
</tr>
<tr>
<td>VCS0 5</td>
<td>Context Load Protocol Register</td>
<td>0b</td>
</tr>
<tr>
<td>VCS0 4</td>
<td>Context Load Protocol Register</td>
<td>0b</td>
</tr>
</tbody>
</table>
### MFX0_CTX_LD_PRTCL - Context Load Protocol Register VCS0

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Context Load Protocol Register - VCS0 0</td>
<td>0b</td>
<td>R/W</td>
<td>This bit is self clear.</td>
</tr>
<tr>
<td>1</td>
<td>Context Load Protocol Register - VCS0 1</td>
<td>0b</td>
<td>R/W</td>
<td>Bit 1 = Context Launched. This bit is self clear.</td>
</tr>
<tr>
<td>2</td>
<td>Context Load Protocol Register - VCS0 2</td>
<td>0b</td>
<td>R/W</td>
<td>Context Load Protocol Register (Written by VCS0) Bit 2 = Request from VCS0 to GAM for context save readiness. GAM will acknowledge appropriated steps taken. This bit is self clear.</td>
</tr>
<tr>
<td>3</td>
<td>Context Load Protocol Register - VCS0 3</td>
<td>0b</td>
<td>R/W</td>
<td>Bit 3 = Context Complete. Only valid with Bit 2 asserted and indicates whether the context switching out as complete vs. incomplete. This bit is self clear.</td>
</tr>
</tbody>
</table>
## Context Load Protocol Register VCS1

<table>
<thead>
<tr>
<th>MFX1_CTX_LD_PRTCL - Context Load Protocol Register VCS1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td><strong>Mask Bits</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td><strong>Context Load Protocol Register - VCS1 15</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For Future Use. This bit is self clear.</td>
</tr>
<tr>
<td>14</td>
<td></td>
<td><strong>Context Load Protocol Register - VCS1 14</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For Future Use. This bit is self clear.</td>
</tr>
<tr>
<td>13</td>
<td></td>
<td><strong>Context Load Protocol Register - VCS1 13</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For Future Use. This bit is self clear.</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td><strong>Context Load Protocol Register - VCS1 12</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For Future Use. This bit is self clear.</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td><strong>Context Load Protocol Register - VCS1 11</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For Future Use. This bit is self clear.</td>
</tr>
</tbody>
</table>
## MFX1_CTX_LD_PRTCL - Context Load Protocol Register VCS1

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>Context Load Protocol Register - VCS1 10</td>
<td>0b</td>
<td>R/W</td>
<td>For Future Use. This bit is self clear.</td>
</tr>
<tr>
<td>9</td>
<td>Context Load Protocol Register - VCS1 9</td>
<td>0b</td>
<td>R/W</td>
<td>For Future Use. This bit is self clear.</td>
</tr>
<tr>
<td>8</td>
<td>Context Load Protocol Register - VCS1 8</td>
<td>0b</td>
<td>R/W</td>
<td>For Future Use. This bit is self clear.</td>
</tr>
<tr>
<td>7</td>
<td>Context Load Protocol Register - VCS1 7</td>
<td>0b</td>
<td>R/W</td>
<td>For Future Use. This bit is self clear.</td>
</tr>
<tr>
<td>6</td>
<td>Context Load Protocol Register - VCS1 6</td>
<td>0b</td>
<td>R/W</td>
<td>For Future Use. This bit is self clear.</td>
</tr>
<tr>
<td>5</td>
<td>Context Load Protocol Register - VCS1 5</td>
<td>0b</td>
<td>R/W</td>
<td>For Future Use. This bit is self clear.</td>
</tr>
<tr>
<td>4</td>
<td>Context Load Protocol Register - VCS1 4</td>
<td>0b</td>
<td>R/W</td>
<td>For Future Use. This bit is self clear.</td>
</tr>
</tbody>
</table>
### MFX1_CTX_LD_PRTCL - Context Load Protocol Register VCS1

<table>
<thead>
<tr>
<th></th>
<th>Context Load Protocol Register - VCS1 3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td>Bit 3 = Context Complete. Only valid with Bit 2 asserted and indicates whether the context switching out as complete vs. incomplete. This bit is self clear.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Context Load Protocol Register - VCS1 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td>Context Load Protocol Register (Written by VCS1)</td>
</tr>
<tr>
<td></td>
<td>Bit 2 = Request from VCS1 to GAM for context save readiness. GAM will acknowledge appropriated steps taken. This bit is self clear.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Context Load Protocol Register - VCS1 1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td>Context Load Protocol Register (Written by VCS1)</td>
</tr>
<tr>
<td></td>
<td>Bit 1 = Context Launched</td>
</tr>
<tr>
<td></td>
<td>This bit is self clear.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Context Load Protocol Register - VCS1 0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td>Context Load Protocol Register (Written by VCS1)</td>
</tr>
<tr>
<td></td>
<td>Bit 0 = Context Available.</td>
</tr>
<tr>
<td></td>
<td>This bit is self clear.</td>
</tr>
</tbody>
</table>
## Context Load Protocol Register VEBX

<table>
<thead>
<tr>
<th>VEBX_CTX_LD_PRTCL - Context Load Protocol Register VEBX</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000 CHV, BSW</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 04010h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td><strong>Mask Bits</strong></td>
</tr>
<tr>
<td></td>
<td>Default Value: 0000h</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access: RO</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td></td>
<td><strong>Context Load Protocol Register - VEBX 15</strong></td>
</tr>
<tr>
<td></td>
<td>Default Value: 0b</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td>For Future Use. This bit is self clear.</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td><strong>Context Load Protocol Register - VEBX 14</strong></td>
</tr>
<tr>
<td></td>
<td>Default Value: 0b</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td>For Future Use. This bit is self clear.</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
<td><strong>Context Load Protocol Register - VEBX 13</strong></td>
</tr>
<tr>
<td></td>
<td>Default Value: 0b</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td>For Future Use. This bit is self clear.</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td><strong>Context Load Protocol Register - VEBX 12</strong></td>
</tr>
<tr>
<td></td>
<td>Default Value: 0b</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td>For Future Use. This bit is self clear.</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td><strong>Context Load Protocol Register - VEBX 11</strong></td>
</tr>
<tr>
<td></td>
<td>Default Value: 0b</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td>For Future Use. This bit is self clear.</td>
<td></td>
</tr>
<tr>
<td>Address</td>
<td>Description</td>
<td>Default Value</td>
</tr>
<tr>
<td>---------</td>
<td>----------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>10</td>
<td>Context Load Protocol Register -</td>
<td>0b</td>
</tr>
<tr>
<td></td>
<td>VEBX 10</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Context Load Protocol Register -</td>
<td>0b</td>
</tr>
<tr>
<td></td>
<td>VEBX 9</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Context Load Protocol Register -</td>
<td>0b</td>
</tr>
<tr>
<td></td>
<td>VEBX 8</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Context Load Protocol Register -</td>
<td>0b</td>
</tr>
<tr>
<td></td>
<td>VEBX 7</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Context Load Protocol Register -</td>
<td>0b</td>
</tr>
<tr>
<td></td>
<td>VEBX 6</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Context Load Protocol Register -</td>
<td>0b</td>
</tr>
<tr>
<td></td>
<td>VEBX 5</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Context Load Protocol Register -</td>
<td>0b</td>
</tr>
<tr>
<td></td>
<td>VEBX 4</td>
<td></td>
</tr>
</tbody>
</table>
**VEBX_CTX_LD_PRTCL - Context Load Protocol Register VEBX**

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Context Load Protocol Register - VEBX 3</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>Default Value: 0b</td>
<td>Project: CHV, BSW</td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td>Bit 3 = Context Complete. Only valid with Bit 2 asserted and indicates whether the context switching out as complete vs. incomplete. This bit is self clear.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Context Load Protocol Register - VEBX 2</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>Default Value: 0b</td>
<td>Access: R/W</td>
<td>Context Load Protocol Register (Written by VEBX)</td>
</tr>
<tr>
<td></td>
<td>Bit 2 = Request from VEBX to GAM for context save readiness. GAM will acknowledge appropriated steps taken. This bit is self clear.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Context Load Protocol Register - VEBX 1</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>Default Value: 0b</td>
<td>Access: R/W</td>
<td>Context Load Protocol Register (Written by VEBX)</td>
</tr>
<tr>
<td></td>
<td>Bit 1 = Context Launched. This bit is self clear.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>Context Load Protocol Register - VEBX 0</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>Default Value: 0b</td>
<td>Access: R/W</td>
<td>Context Load Protocol Register (Written by VEBX)</td>
</tr>
<tr>
<td></td>
<td>Bit 0 = Context Available. This bit is self clear.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Context Restore Request To TDL

<table>
<thead>
<tr>
<th>TDL_CONTEXT_RESTORE - Context Restore Request To TDL</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Access:</strong> WO</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 0E440h</td>
</tr>
<tr>
<td><strong>Valid Projects:</strong> CHV, BSW</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:17</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td><strong>Context Restore Mask</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Value</strong></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>15:1</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td><strong>Context Restore</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Value</strong></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>
## Context Save Request To TDL

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>WO</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0E4FCh</td>
</tr>
<tr>
<td>Valid Projects:</td>
<td>CHV, BSW</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:17</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format:</strong> MBZ</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td><strong>Context Save Mask</strong></td>
</tr>
<tr>
<td></td>
<td>Value</td>
<td>Name</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Bit 0 and Bit 16 both need to be '1' for Context Save Request</td>
</tr>
<tr>
<td>15:1</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format:</strong> MBZ</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td><strong>Context Save</strong></td>
</tr>
<tr>
<td></td>
<td>Value</td>
<td>Name</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Bit 0 and Bit 16 both need to be '1' for Context Save Request</td>
</tr>
</tbody>
</table>
## Context Sizes

### CXT_SIZE - Context Sizes

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>RenderCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x05655582</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Trusted Type:</td>
<td>1</td>
</tr>
<tr>
<td>Address:</td>
<td>021A8h</td>
</tr>
</tbody>
</table>

The actual size of a logical rendering context is the amount of data stored/restored during a context switch and is measured in 64B cache lines.

This register will be power context save/restored. Note that this register will default to the correct value, so software should not have to modify it.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:27</td>
<td><strong>Reserved</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>26:24</td>
<td>Ring Context Size</td>
<td>This field indicates the Ring context data that needs to be save/restored.</td>
<td>5h</td>
<td></td>
</tr>
<tr>
<td>23:16</td>
<td>Render Context Size</td>
<td>This field indicates the size of the render context data that needs to be save/restored when extended mode is not enabled for a context; this also excludes VF, VFE, and URB context size.</td>
<td>65h</td>
<td></td>
</tr>
<tr>
<td>15:8</td>
<td>SOL Context Offset</td>
<td>This field indicates the cacheline aligned offset of the SOL context in the render context image starting from Ring Context. Note that in execlist of scheduling Ring context itself is at 4KB offset from LRCA.</td>
<td>55h</td>
<td></td>
</tr>
<tr>
<td>7:0</td>
<td>VF and VFE State Context Size</td>
<td>This field indicates the amount of VF and VFE unit data context save/restored in cachelines.</td>
<td>82h</td>
<td></td>
</tr>
</tbody>
</table>
## Context Status Buffer Contents

<table>
<thead>
<tr>
<th>CTXT_ST_BUF - Context Status Buffer Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value:</td>
</tr>
<tr>
<td>0x00000000, 0x00000000, 0x00000000, 0x00000000,</td>
</tr>
<tr>
<td>0x00000000, 0x00000000, 0x00000000, 0x00000000,</td>
</tr>
<tr>
<td>0x00000000, 0x00000000, 0x00000000, 0x00000000</td>
</tr>
<tr>
<td>Access: R/W</td>
</tr>
<tr>
<td>Size (in bits): 384</td>
</tr>
<tr>
<td>Trusted Type: 1</td>
</tr>
<tr>
<td>Address: 02370h-0239Fh</td>
</tr>
<tr>
<td>Name: Context Status Buffer Contents</td>
</tr>
<tr>
<td>ShortName: CTXT_ST_BUF_RCSUNIT</td>
</tr>
<tr>
<td>Address: 12370h-1239Fh</td>
</tr>
<tr>
<td>Name: Context Status Buffer Contents</td>
</tr>
<tr>
<td>ShortName: CTXT_ST_BUF_VCSUNIT0</td>
</tr>
<tr>
<td>Address: 1A370h-1A39Fh</td>
</tr>
<tr>
<td>Name: Context Status Buffer Contents</td>
</tr>
<tr>
<td>ShortName: CTXT_ST_BUF_VECSUNIT</td>
</tr>
<tr>
<td>Address: 1C370h-1C39Fh</td>
</tr>
<tr>
<td>Name: Context Status Buffer Contents</td>
</tr>
<tr>
<td>ShortName: CTXT_ST_BUF_VCSUNIT1</td>
</tr>
<tr>
<td>Address: 22370h-2239Fh</td>
</tr>
<tr>
<td>Name: Context Status Buffer Contents</td>
</tr>
<tr>
<td>ShortName: CTXT_ST_BUF_BCSUNIT</td>
</tr>
</tbody>
</table>

Contents of the Execlist 0 in HW.

### Programming Notes

This structure contains the Context Switch status locations Context Status 0 to Context Status 5.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>63:32</td>
<td><strong>Context Status 0 UDW</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: Context Status CHV, BSW</td>
</tr>
<tr>
<td></td>
<td>31:0</td>
<td><strong>Context Status 0 LDW</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: Context Status CHV, BSW</td>
</tr>
<tr>
<td>1</td>
<td>63:32</td>
<td><strong>Context Status 1 UDW</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: Context Status CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Context Status Buffer Contents</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>--------------------------------</td>
</tr>
</tbody>
</table>
|   | 31:0 | **Context Status 1 LDW**  
Format: Context Status CHV, BSW |
| 2 | 63:32 | **Context Status 2 UDW**  
Format: Context Status CHV, BSW |
|   | 31:0 | **Context Status 2 LDW**  
Format: Context Status CHV, BSW |
| 3 | 63:32 | **Context Status 3 UDW**  
Format: Context Status CHV, BSW |
|   | 31:0 | **Context Status 3 LDW**  
Format: Context Status CHV, BSW |
| 4 | 63:32 | **Context Status 4 UDW**  
Format: Context Status CHV, BSW |
|   | 31:0 | **Context Status 4 LDW**  
Format: Context Status CHV, BSW |
| 5 | 63:32 | **Context Status 5 UDW**  
Format: Context Status CHV, BSW |
|   | 31:0 | **Context Status 5 LDW**  
Format: Context Status CHV, BSW |
## Control Register for Power Management

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>31:16</td>
<td>Mask Bits</td>
<td>Format: Mask[15:0] Must be set to modify corresponding bit in Bits 15:0. (All implemented bits)</td>
</tr>
<tr>
<td>15:14</td>
<td>Reserved</td>
<td>Format: MBZ</td>
</tr>
</tbody>
</table>

**WAIT_FOR_RC6_EXIT - Control Register for Power Management**

- **Register Space:** MMIO: 0/2/0
- **Project:** CHV, BSW
- **Source:** RenderCS
- **Default Value:** 0x00000000
- **Access:** R/W
- **Size (in bits):** 32
- **Trusted Type:** 1

**Address:**
- 020CCh
- 120CCh-120CFh
- 1A0CCh-1A0CFh
- 1C0CCh-1C0CFh
- 220CCh-220CFh

**Name:** Control Register for Power Management

**ShortName:**
- WAIT_FOR_RC6_EXIT_VCSUNIT0
- WAIT_FOR_RC6_EXIT_VECSUNIT
- WAIT_FOR_RC6_EXIT_VCSUNIT1
- WAIT_FOR_RC6_EXIT_BCSUNIT
**WAIT_FOR_RC6_EXIT - Control Register for Power Management**

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Get power context save/restored. Bit[0] contents of this register doesn't get save/restored.</td>
</tr>
<tr>
<td>13</td>
<td><strong>Selective Read Addressing Enable</strong></td>
</tr>
<tr>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td>This field controls the outbound read request originating from Render Command Streamer. This field enables to read the MMIO register from selected unit in a given slice and sub-slice instead of multicasting the read cycle to all slices/sub-slices.</td>
</tr>
<tr>
<td></td>
<td><strong>Value</strong></td>
</tr>
<tr>
<td>0h</td>
<td>[Default]</td>
</tr>
<tr>
<td>1h</td>
<td>Selective Unit Enabled</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>12:11</td>
<td><strong>Selective Read Slice Select</strong></td>
</tr>
<tr>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td>This field selects the slice from which the read return data value has to be considered when Selective Read Addressing Enable is set. Below value must be programmed with a legal value supported by the GT configuration, one must not program this to a slice that is disabled or not supported by GT.</td>
</tr>
<tr>
<td></td>
<td><strong>Value</strong></td>
</tr>
<tr>
<td>00b</td>
<td>Slice-0</td>
</tr>
<tr>
<td>01b</td>
<td>Slice-1</td>
</tr>
<tr>
<td>10b</td>
<td>Slice-2</td>
</tr>
<tr>
<td>11b</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>10:9</td>
<td><strong>Selective Read Sub-Slice Select</strong></td>
</tr>
<tr>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td>This field selects the sub-slice from which the read return data value has to be considered when Selective Read Addressing Enable is set. Below value must be programmed with a legal value supported by the GT configuration, one must not program this to a sub-slice that is disabled or not supported by GT.</td>
</tr>
<tr>
<td></td>
<td><strong>Value</strong></td>
</tr>
<tr>
<td>00b</td>
<td>Sub Slice-0</td>
</tr>
<tr>
<td>01b</td>
<td>Sub Slice-1</td>
</tr>
<tr>
<td>10b</td>
<td>Sub Slice-2</td>
</tr>
<tr>
<td>11b</td>
<td>Reserved</td>
</tr>
</tbody>
</table>
## WAIT_FOR_RC6_EXIT - Control Register for Power Management

### 8 Render Inhibit

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>Disabled</td>
<td>When not Set CS doesn't take any special action.</td>
</tr>
<tr>
<td></td>
<td>[Default]</td>
<td></td>
</tr>
<tr>
<td>1h</td>
<td>Enabled</td>
<td>When Set CS will not save/restore render context as part of power context save/restore. Render context includes RS context as well if enabled.</td>
</tr>
</tbody>
</table>

**Programming Notes**

If this bit is set S/W should set Resource Streamer Context Enable (Bit[7] of this register) as well.

### 7 Resource Streamer Context Enable

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1h</td>
<td>Disable</td>
<td>The current context does not include the resource streamer context</td>
</tr>
<tr>
<td>0h</td>
<td>Enable</td>
<td>The current context does include the resource streamer context.</td>
</tr>
<tr>
<td></td>
<td>[Default]</td>
<td></td>
</tr>
</tbody>
</table>

### 6 Selective Write Addressing Enable

- **Project:** CHV, BSW
- **Description:** This field controls the outbound write request on message channel originating from Render Command Streamer on executing `MI_LOAD_REGISTER_IMM`, `MI_LOAD_REGISTER_REG` and `MI_LOAD_REGISTER_MEM` commands. Setting this field doesn't affect the execution of `MI_LOAD_REGISTER_IMM` command from context image during context restore. This field enables to direct the message channel write cycle to the unit in the selected slice and sub-slice instead of multicasting it to all the instances of the unit in all the slices and sub-slices.

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>Multi Cast</td>
<td>[Default]</td>
</tr>
<tr>
<td>1h</td>
<td>Selective Unit Enabled</td>
<td>Unit selected based on Selective Write Slice Select and Selective Write Sub-Slice Select.</td>
</tr>
</tbody>
</table>
WAIT_FOR_RC6_EXIT - Control Register for Power Management

5:4 Selective Write Slice Select

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>00b</td>
<td>Slice-0</td>
</tr>
<tr>
<td>01b</td>
<td>Slice-1</td>
</tr>
<tr>
<td>10b</td>
<td>Slice-2</td>
</tr>
<tr>
<td>11b</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

This field selects the slice to which the write has to be done when Selective Write Addressing Enable is set. Below value must be programmed with a legal value supported by the GT configuration, one must not program this to a slice that is disabled or not supported by GT.

3:2 Selective Write Sub-Slice Select

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>00b</td>
<td>Sub Slice-0</td>
</tr>
<tr>
<td>01b</td>
<td>Sub Slice-1</td>
</tr>
<tr>
<td>10b</td>
<td>Sub Slice-2</td>
</tr>
<tr>
<td>11b</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

This field selects the Sub-Slice to which the write has to be done when Selective Write Addressing Enable is set. Below value must be programmed with a legal value supported by the GT configuration, one must not program this to a sub-slice that is disabled or not supported by GT.

1 Reserved

Format: MBZ

0 WAIT FOR RC6 EXIT

Format: Disable

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>Disabled [Default]</td>
<td>When not Set CS doesn't take any action.</td>
</tr>
<tr>
<td>1h</td>
<td>Enabled</td>
<td>When Set CS will stop on the next appropriate command boundary and will initiate IDLE sequence with PM.</td>
</tr>
</tbody>
</table>

Programming Notes

WAIT_FOR_RC6_EXIT functionality is only supported in ring buffer mode of scheduling and not supported in execlist mode of scheduling.
Count Active Channels Dispatched

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>63:32</td>
<td><strong>GPGPU_THREADS_DISPATCHED UDW</strong></td>
</tr>
<tr>
<td></td>
<td>Format:</td>
<td>U32</td>
</tr>
<tr>
<td></td>
<td>This count is increased by the number of active bits in the execution mask each time the TS sends a GPGPU dispatch.</td>
<td></td>
</tr>
<tr>
<td>31:0</td>
<td></td>
<td><strong>GPGPU_THREADS_DISPATCHED LDW</strong></td>
</tr>
<tr>
<td></td>
<td>Format:</td>
<td>U32</td>
</tr>
<tr>
<td></td>
<td>This count is increased by the number of active bits in the execution mask each time the TS sends a GPGPU dispatch.</td>
<td></td>
</tr>
</tbody>
</table>
CS Context Timestamp Count

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>RenderCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>023A8h</td>
</tr>
</tbody>
</table>

This register provides a mechanism to obtain cumulative run time of a GPU context on HW. This register gets context save/restored on a context switch. SW must reset this register on very first submission of a context to HW, then afterwards gets context save/restored maintaining the cumulative run time of the corresponding context. This register (effectively) counts at a constant frequency by adjusting the increment amount according to the actual reference clock frequency. SW therefore does not need to know the reference clock frequency. This register gets reset on an engine reset.

This register is context save restore on a context switch. The time to execute the context switch is included in the CS_CTX_TIMESTAMP register.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Timestamp Value</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: U32</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This register increments for every 80 ns of time.</td>
</tr>
</tbody>
</table>
## CS General Purpose Register

<table>
<thead>
<tr>
<th>CS_GPR - CS General Purpose Register</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> RenderCS</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000, 0x00000000</td>
</tr>
<tr>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 64</td>
</tr>
<tr>
<td><strong>Address:</strong> 02600h</td>
</tr>
<tr>
<td><strong>Name:</strong> CS General Purpose Register 0</td>
</tr>
<tr>
<td><strong>ShortName:</strong> CS_GPR_R_0</td>
</tr>
<tr>
<td><strong>Address:</strong> 02608h</td>
</tr>
<tr>
<td><strong>Name:</strong> CS General Purpose Register 1</td>
</tr>
<tr>
<td><strong>ShortName:</strong> CS_GPR_R_1</td>
</tr>
<tr>
<td><strong>Address:</strong> 02610h</td>
</tr>
<tr>
<td><strong>Name:</strong> CS General Purpose Register 2</td>
</tr>
<tr>
<td><strong>ShortName:</strong> CS_GPR_R_2</td>
</tr>
<tr>
<td><strong>Address:</strong> 02618h</td>
</tr>
<tr>
<td><strong>Name:</strong> CS General Purpose Register 3</td>
</tr>
<tr>
<td><strong>ShortName:</strong> CS_GPR_R_3</td>
</tr>
<tr>
<td><strong>Address:</strong> 02620h</td>
</tr>
<tr>
<td><strong>Name:</strong> CS General Purpose Register 4</td>
</tr>
<tr>
<td><strong>ShortName:</strong> CS_GPR_R_4</td>
</tr>
<tr>
<td><strong>Address:</strong> 02628h</td>
</tr>
<tr>
<td><strong>Name:</strong> CS General Purpose Register 5</td>
</tr>
<tr>
<td><strong>ShortName:</strong> CS_GPR_R_5</td>
</tr>
<tr>
<td><strong>Address:</strong> 02630h</td>
</tr>
<tr>
<td><strong>Name:</strong> CS General Purpose Register 6</td>
</tr>
<tr>
<td><strong>ShortName:</strong> CS_GPR_R_6</td>
</tr>
<tr>
<td><strong>Address:</strong> 02638h</td>
</tr>
<tr>
<td><strong>Name:</strong> CS General Purpose Register 7</td>
</tr>
<tr>
<td><strong>ShortName:</strong> CS_GPR_R_7</td>
</tr>
<tr>
<td><strong>Address:</strong> 02640h</td>
</tr>
<tr>
<td><strong>Name:</strong> CS General Purpose Register 8</td>
</tr>
<tr>
<td><strong>ShortName:</strong> CS_GPR_R_8</td>
</tr>
</tbody>
</table>
### CS_GPR - CS General Purpose Register

<table>
<thead>
<tr>
<th>Address</th>
<th>02648h</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name:</td>
<td>CS General Purpose Register 9</td>
</tr>
<tr>
<td>ShortName:</td>
<td>CS_GPR_R_9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Address</th>
<th>02650h</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name:</td>
<td>CS General Purpose Register 10</td>
</tr>
<tr>
<td>ShortName:</td>
<td>CS_GPR_R_10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Address</th>
<th>02658h</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name:</td>
<td>CS General Purpose Register 11</td>
</tr>
<tr>
<td>ShortName:</td>
<td>CS_GPR_R_11</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Address</th>
<th>02660h</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name:</td>
<td>CS General Purpose Register 12</td>
</tr>
<tr>
<td>ShortName:</td>
<td>CS_GPR_R_12</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Address</th>
<th>02668h</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name:</td>
<td>CS General Purpose Register 13</td>
</tr>
<tr>
<td>ShortName:</td>
<td>CS_GPR_R_13</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Address</th>
<th>02670h</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name:</td>
<td>CS General Purpose Register 14</td>
</tr>
<tr>
<td>ShortName:</td>
<td>CS_GPR_R_14</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Address</th>
<th>02678h</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name:</td>
<td>CS General Purpose Register 15</td>
</tr>
<tr>
<td>ShortName:</td>
<td>CS_GPR_R_15</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Address</th>
<th>12600h-12607h</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name:</td>
<td>General Purpose Register</td>
</tr>
<tr>
<td>ShortName:</td>
<td>CS_GPR_VCSUNIT0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Address</th>
<th>1A600h-1A607h</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name:</td>
<td>General Purpose Register</td>
</tr>
<tr>
<td>ShortName:</td>
<td>CS_GPR_VECSUNIT</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Address</th>
<th>1C600h-1C607h</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name:</td>
<td>General Purpose Register</td>
</tr>
<tr>
<td>ShortName:</td>
<td>CS_GPR_VCSUNIT1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Address</th>
<th>22600h-22607h</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name:</td>
<td>General Purpose Register</td>
</tr>
<tr>
<td>ShortName:</td>
<td>CS_GPR_BCSUNIT</td>
</tr>
</tbody>
</table>

This is a General Purpose Register bank of sixteen 64bit registers, which will be used as temporary storage by MI_MATH command to do ALU operations.
## CS_GPR - CS General Purpose Register

<table>
<thead>
<tr>
<th>GPR Index</th>
<th>MMIO Offset</th>
</tr>
</thead>
<tbody>
<tr>
<td>R_0</td>
<td>0x2600</td>
</tr>
<tr>
<td>R_1</td>
<td>0x2608</td>
</tr>
<tr>
<td>R_2</td>
<td>0x2610</td>
</tr>
<tr>
<td>R_3</td>
<td>0x2618</td>
</tr>
<tr>
<td>R_4</td>
<td>0x2620</td>
</tr>
<tr>
<td>R_5</td>
<td>0x2628</td>
</tr>
<tr>
<td>R_6</td>
<td>0x2630</td>
</tr>
<tr>
<td>R_7</td>
<td>0x2638</td>
</tr>
<tr>
<td>R_8</td>
<td>0x2640</td>
</tr>
<tr>
<td>R_9</td>
<td>0x2648</td>
</tr>
<tr>
<td>R_10</td>
<td>0x2650</td>
</tr>
<tr>
<td>R_11</td>
<td>0x2658</td>
</tr>
<tr>
<td>R_12</td>
<td>0x2660</td>
</tr>
<tr>
<td>R_13</td>
<td>0x2668</td>
</tr>
<tr>
<td>R_14</td>
<td>0x2670</td>
</tr>
<tr>
<td>R_15</td>
<td>0x2678</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>63:0</td>
<td><strong>CS_GPR_DATA</strong></td>
</tr>
</tbody>
</table>

- **Project:** CHV, BSW

This register is a temporary register for ALU operations. See MI_MATH command for more details.
## CS Power Management FSM

<table>
<thead>
<tr>
<th>CSPWRFSM - CS Power Management FSM</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> RenderCS</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 022ACh</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Value</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:30</td>
<td>Reserved</td>
<td>Format: MBZ</td>
<td></td>
</tr>
<tr>
<td>29:28</td>
<td>CSFBCSLICE0</td>
<td>FBC message forward FSM state</td>
<td>Format: U2</td>
<td></td>
</tr>
<tr>
<td>27:24</td>
<td>Reserved</td>
<td>Format: MBZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23:21</td>
<td>CS ARB</td>
<td>Overall state of the command streamer. Describes what state CS is in</td>
<td>Format: U3</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>CSFBCIDLE_0</td>
<td></td>
</tr>
<tr>
<td>1h</td>
<td>CSFBCMODIFY_0</td>
<td></td>
</tr>
<tr>
<td>2h</td>
<td>CSFBCCLEAN_0</td>
<td></td>
</tr>
<tr>
<td>3h</td>
<td>CSFBCDONE_0</td>
<td></td>
</tr>
<tr>
<td>0h</td>
<td>ARBIDLE_s</td>
<td></td>
</tr>
<tr>
<td>1h</td>
<td>P0RNG_s CS</td>
<td></td>
</tr>
<tr>
<td>2h</td>
<td>P0BATCH_s</td>
<td></td>
</tr>
<tr>
<td>3h</td>
<td>ARBCHK</td>
<td></td>
</tr>
<tr>
<td>4h</td>
<td>ARBCHK1</td>
<td></td>
</tr>
<tr>
<td>5h</td>
<td>CTXOP_s</td>
<td></td>
</tr>
<tr>
<td>6h</td>
<td>WABATCH_s</td>
<td></td>
</tr>
<tr>
<td>7h</td>
<td>PSLBATCH</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>ARBIDLE_s</td>
<td></td>
</tr>
<tr>
<td>1h</td>
<td>P0RNG_s CS</td>
<td></td>
</tr>
<tr>
<td>2h</td>
<td>P0BATCH_s</td>
<td></td>
</tr>
<tr>
<td>3h</td>
<td>ARBCHK</td>
<td></td>
</tr>
<tr>
<td>4h</td>
<td>ARBCHK1</td>
<td></td>
</tr>
<tr>
<td>5h</td>
<td>CTXOP_s</td>
<td></td>
</tr>
<tr>
<td>6h</td>
<td>WABATCH_s</td>
<td></td>
</tr>
<tr>
<td>7h</td>
<td>PSLBATCH</td>
<td></td>
</tr>
</tbody>
</table>

CHV, BSW
### CSPWRFSM - CS Power Management FSM

<table>
<thead>
<tr>
<th>Register</th>
<th>Field</th>
<th>Mask</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>Reserved</td>
<td></td>
<td>MBZ</td>
<td></td>
</tr>
</tbody>
</table>

#### CSSWITCH

- **Field**: 19:17
- **Format**: U3
- **Description**: Arbiters CSSWITCH FSM state decoding.

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>SWIDLE_s</td>
</tr>
<tr>
<td>1h</td>
<td>SWITCH_s</td>
</tr>
<tr>
<td>2h</td>
<td>ASREQ_s</td>
</tr>
<tr>
<td>3h</td>
<td>DMACHK_s</td>
</tr>
<tr>
<td>4h</td>
<td>ARBWAIT_s</td>
</tr>
<tr>
<td>5h</td>
<td>FIFORECFG_s</td>
</tr>
<tr>
<td>6h-7h</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

#### CSCSBUPDATE

- **Field**: 16:13
- **Format**: U4
- **Description**: CS Power Management CSBLOCK FSM state

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>CSBIDLE</td>
</tr>
<tr>
<td>1h</td>
<td>CSQ</td>
</tr>
<tr>
<td>2h</td>
<td>WRPTR</td>
</tr>
<tr>
<td>3h</td>
<td>SEMA1</td>
</tr>
<tr>
<td>4h</td>
<td>SEMA2</td>
</tr>
<tr>
<td>5h</td>
<td>TS1</td>
</tr>
<tr>
<td>6h</td>
<td>TS2</td>
</tr>
<tr>
<td>7h</td>
<td>TS3</td>
</tr>
<tr>
<td>8h</td>
<td>TS4</td>
</tr>
<tr>
<td>9h</td>
<td>DUMMYREQ</td>
</tr>
<tr>
<td>Ah</td>
<td>DUMMYWT</td>
</tr>
<tr>
<td>Bh</td>
<td>INTWT</td>
</tr>
<tr>
<td>Ch-Fh</td>
<td>Reserved</td>
</tr>
</tbody>
</table>
## CSPWRFSM - CS Power Management FSM

### 12:11 R2MWRREQ

**Format:** U2

CSSTDT memory request FSM state

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>WRIDLE</td>
</tr>
<tr>
<td>1h</td>
<td>WRREQ_HW1</td>
</tr>
<tr>
<td>2h</td>
<td>WRREQ_HW2</td>
</tr>
<tr>
<td>3h</td>
<td>WRRD</td>
</tr>
</tbody>
</table>

### 10 Reserved

**Format:** MBZ

### 9.7 LOADARB

**Format:** U3

CSSTDT arbiter FSM state

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>LDIDLE</td>
</tr>
<tr>
<td>1h</td>
<td>LDAUTO</td>
</tr>
<tr>
<td>2h</td>
<td>LDPRSR</td>
</tr>
<tr>
<td>3h</td>
<td>LDCTX</td>
</tr>
<tr>
<td>4h</td>
<td>LDFLSH</td>
</tr>
<tr>
<td>5h</td>
<td>LDREG</td>
</tr>
<tr>
<td>6h</td>
<td>LDSHR1</td>
</tr>
</tbody>
</table>

**Programming Notes**

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

### 6.4 CSBLOCK

**Format:** U3

CS Power Management CSBLOCK FSM state

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>CSBLOCK</td>
</tr>
<tr>
<td>1h</td>
<td>CSCTXARB</td>
</tr>
<tr>
<td>2h</td>
<td>CSUNBLOCKRESTORE</td>
</tr>
<tr>
<td>3h</td>
<td>CSUNBLOCK</td>
</tr>
<tr>
<td>4h</td>
<td>CSPREP4BLOCK</td>
</tr>
<tr>
<td>5h-7h</td>
<td>Reserved</td>
</tr>
</tbody>
</table>
## CSPWRFSM - CS Power Management FSM

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>CSBUSY</td>
</tr>
<tr>
<td>1h</td>
<td>CNTWT</td>
</tr>
<tr>
<td>2h</td>
<td>FLSHREQ</td>
</tr>
<tr>
<td>3h</td>
<td>FLSHWT</td>
</tr>
<tr>
<td>4h</td>
<td>CTXSAVE</td>
</tr>
<tr>
<td>5h</td>
<td>CSREQBLOCK</td>
</tr>
<tr>
<td>6h</td>
<td>PMTURNOFF</td>
</tr>
<tr>
<td>7h</td>
<td>PMIDLEWT</td>
</tr>
<tr>
<td>8h</td>
<td>IDLE</td>
</tr>
<tr>
<td>9h</td>
<td>PMTURNON</td>
</tr>
<tr>
<td>Ah</td>
<td>PMBUSYWT</td>
</tr>
<tr>
<td>8h</td>
<td>DOPFFCGREQ</td>
</tr>
<tr>
<td>Ch</td>
<td>DOPFFCGWAIT</td>
</tr>
<tr>
<td>Dh</td>
<td>DOPFFCG</td>
</tr>
<tr>
<td>Eh</td>
<td>DOPFFCUGREQ</td>
</tr>
<tr>
<td>Fh</td>
<td>DOPFFCUGWAIT</td>
</tr>
</tbody>
</table>
### CSPREEMPT - CSPREEMPT

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>RenderCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Trusted Type:</td>
<td>1</td>
</tr>
<tr>
<td>Address:</td>
<td>024B0h</td>
</tr>
</tbody>
</table>

#### Programming Notes
This is for HW internal usage and must not be written by SW.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td><strong>Mask Bits</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format:</strong> Mask[15:0]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Must be set to modify corresponding bit in Bits 15:0. (All implemented bits)</td>
</tr>
<tr>
<td>15:1</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format:</strong> MBZ</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td><strong>Unnamed</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format:</strong> Disable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This is a message bit written by the cross CS in case of GT4-CBR/SFR mode of operation. To set this bit both bit[0] and bit[16] (mask) needs to be set. This bit set indicates CS in other GT has reached a preemption point. This bit gets reset by CS when preemption takes place.</td>
</tr>
</tbody>
</table>
## CS Reset Control Register

<table>
<thead>
<tr>
<th>CS_RESET_CTRL - CS Reset Control Register</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space:</td>
</tr>
<tr>
<td>MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project:</td>
</tr>
<tr>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
</tr>
<tr>
<td>RenderCS</td>
</tr>
<tr>
<td>Default Value:</td>
</tr>
<tr>
<td>0x0000000000</td>
</tr>
<tr>
<td>Access:</td>
</tr>
<tr>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
</tr>
<tr>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
</tr>
<tr>
<td>020D0h</td>
</tr>
</tbody>
</table>

This register is to be used to control soft reset.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td><strong>Mask Bits</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: Mask[15:0]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Must be set to modify corresponding bit in Bits 15:0. (All implemented bits)</td>
</tr>
<tr>
<td>15:2</td>
<td>Reserved</td>
<td>Format: MBZ</td>
</tr>
<tr>
<td>1</td>
<td><strong>Ready for Reset</strong></td>
<td>Format: U1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>When set indicates render engine is ready for reset. This bit gets cleared on engine reset or when Soft Reset In progress is cleared.</td>
</tr>
<tr>
<td>0</td>
<td><strong>Request Reset</strong></td>
<td>Format: U1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>When set indicates SW wishes to reset the render engine. On seeing this bit set Command Streamer will take appropriate action and set Ready For Reset status bit. This bit gets cleared on engine reset.</td>
</tr>
</tbody>
</table>
## Current Context Register

**CCID - Current Context Register**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>RenderCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
</tbody>
</table>

**Address:** 02180h

### Address: 12180h-12183h

<table>
<thead>
<tr>
<th>Name:</th>
<th>Current Context Register</th>
</tr>
</thead>
<tbody>
<tr>
<td>ShortName:</td>
<td>CCID_VCSUNIT0</td>
</tr>
</tbody>
</table>

### Address: 1A180h-1A183h

<table>
<thead>
<tr>
<th>Name:</th>
<th>Current Context Register</th>
</tr>
</thead>
<tbody>
<tr>
<td>ShortName:</td>
<td>CCID_VECSUNIT</td>
</tr>
</tbody>
</table>

### Address: 1C180h-1C183h

<table>
<thead>
<tr>
<th>Name:</th>
<th>Current Context Register</th>
</tr>
</thead>
<tbody>
<tr>
<td>ShortName:</td>
<td>CCID_VCSUNIT1</td>
</tr>
</tbody>
</table>

### Address: 22180h-22183h

<table>
<thead>
<tr>
<th>Name:</th>
<th>Current Context Register</th>
</tr>
</thead>
<tbody>
<tr>
<td>ShortName:</td>
<td>CCID_BCSUNIT</td>
</tr>
</tbody>
</table>

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

### Programming Notes

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:12</td>
<td><strong>Logical Render Context Address (LRCA)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: GraphicsAddress[31:12]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This field contains the 4 KB-aligned Graphics Memory Address of the current Logical Rendering Context. Bit 11 MBZ.</td>
</tr>
</tbody>
</table>

|       | 11:10 | **Reserved**                        |
|       |      | Format: MBZ                         |
## CCID - Current Context Register

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>Regular Context</td>
<td></td>
</tr>
<tr>
<td>1h</td>
<td>HD DVD Context</td>
<td>Special considerations for TDP allow for higher voltage and frequency.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>Reserved</td>
<td>Must Be One</td>
</tr>
<tr>
<td>7:4</td>
<td>Reserved</td>
<td>MBZ</td>
</tr>
<tr>
<td>3</td>
<td>Extended State Save Enable</td>
<td>If set, the extended state identified in the Logical Context Data section of the Memory Data Formats chapter, is saved as part of switching away from this logical context.</td>
</tr>
<tr>
<td>2</td>
<td>Extended State Restore Enable</td>
<td>If set, the extended state identified in the Logical Context Data section of the Memory Data Formats chapter, was loaded (or restored) as part of switching to this logical context.</td>
</tr>
<tr>
<td>1</td>
<td>Reserved</td>
<td>MBZ</td>
</tr>
<tr>
<td>0</td>
<td>Valid</td>
<td>U1</td>
</tr>
</tbody>
</table>

### Value Name Description

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>Invalid [Default]</td>
<td>The other fields of this register are invalid. A switch away from the context will not invoke a context save operation.</td>
</tr>
<tr>
<td>1h</td>
<td>Valid</td>
<td>The other fields of this register are valid, and a switch from the context will invoke the normal context save/restore operations.</td>
</tr>
</tbody>
</table>
Current Idle/Busy/Avg Count for Freq Down Recommendation

RPCURDN - Current Idle/Busy/Avg Count for Freq Down Recommendation

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0A060h-0A063h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RO</td>
</tr>
<tr>
<td>23:0</td>
<td></td>
<td><strong>Current Busy in Down EI</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reports the current busyness in the down evaluation interval</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 = 0 usec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 = 1.28 usec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 = 2.56 usec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 = 3.84 usec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FF FFFF = 21.474 sec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>pmcr_current_ei_down_busy[23:0]</td>
</tr>
</tbody>
</table>
Current Idle/Busy/Avg Count for Freq Up Recommendation

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0A054h-0A057h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>0</td>
<td>23:0</td>
<td><strong>Current Busy in UP EI</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reports the current busyness in the UP evaluation interval</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 = 0 usec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 = 1.28 usec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 = 2.56 usec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 = 3.84 usec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FF FFFF = 21.474 sec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>pmcr_current_ei_up_busy[23:0]</td>
</tr>
</tbody>
</table>
# Current Time in DOWN EI

**RPCURDNEI - Current Time in DOWN EI**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0A05Ch-0A05Fh</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>Reserved</strong></td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Current_Time_in_Down_EI</strong></td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td>23:0</td>
<td>Reports the current Time in the down evaluation interval</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 = 0 usec</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 = 1.28 usec</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 = 2.56 usec</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 = 3.84 usec</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>FF FFFF = 21.474 sec</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>pmcr_current_ei_down_time[23:0]</td>
<td></td>
</tr>
</tbody>
</table>
# Current Time in UP EI

## RPCURUPEI - Current Time in UP EI

<table>
<thead>
<tr>
<th>Address:</th>
<th>0A050h-0A053h</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>Reserved</strong></td>
<td>RO</td>
</tr>
<tr>
<td>23:0</td>
<td></td>
<td><strong>Current Time</strong></td>
<td>RO</td>
</tr>
</tbody>
</table>

Reports the current time in UP EI

- 0 = 0 usec
- 1 = 1.28 usec
- 2 = 2.56 usec
- 3 = 3.84 usec
- FF FFFF = 21.474 sec

pmcr_current_ei_up_time[23:0]
### CEC0-0 - Customizable Event Creation 0-0

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>02770h</td>
</tr>
<tr>
<td>Valid Projects:</td>
<td>CHV, BSW</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:21</td>
<td>Negate</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format:</strong> U11</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The 11-bit array defined in this field allows input bus bits [10:0] to be individually negated in order to facilitate more complicated custom event creation (e.g. A &amp; (!B</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Value</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1b</td>
</tr>
<tr>
<td>20:19</td>
<td></td>
<td>Source Select</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format:</strong> U2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Selects the input signals to the Boolean event definition logic (see block diagram in the Custom Event Counters section).</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Value</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>01b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>11b</td>
</tr>
<tr>
<td>18:3</td>
<td></td>
<td>Compare Value</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format:</strong> U16</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The value in this field is compared the 16-bit conditioned input bus that are fed into the comparator (see block diagram in the Custom Event Counters section). The type of comparison that is done is controlled by the Compare Function. When the compare function is true, then the signal for the custom event is asserted. This signal in turn can be counted by the B0 performance counter or fed into other CEC blocks.</td>
</tr>
</tbody>
</table>
## CEC0-0 - Customizable Event Creation 0-0

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>000b</td>
<td>Any Are Equal</td>
<td>Compare and assert output if any are equal (Can be used as OR function)</td>
</tr>
<tr>
<td>001b</td>
<td>Greater Than</td>
<td>Compare and assert output if greater than</td>
</tr>
<tr>
<td>010b</td>
<td>Equal</td>
<td>Compare and assert output if equal to (Can also be used as AND function)</td>
</tr>
<tr>
<td>011b</td>
<td>Greater Than or Equal</td>
<td>Compare and assert output if greater than or equal</td>
</tr>
<tr>
<td>100b</td>
<td>Less Than</td>
<td>Compare and assert output if less than</td>
</tr>
<tr>
<td>101b</td>
<td>Not Equal</td>
<td>Compare and assert output if not equal</td>
</tr>
<tr>
<td>110b</td>
<td>Less Than or Equal</td>
<td>Compare and assert output if less than or equal</td>
</tr>
<tr>
<td>111b</td>
<td>Reserved</td>
<td></td>
</tr>
</tbody>
</table>
## Customizable Event Creation 1-0

### CEC1-0 - Customizable Event Creation 1-0

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>02778h</td>
</tr>
<tr>
<td>Valid Projects:</td>
<td>CHV, BSW</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:21</td>
<td><strong>Negate</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format:</strong> U11</td>
</tr>
</tbody>
</table>

The 11-bit array defined in this field allows input bus bits [10:0] to be individually negated in order to facilitate more complicated custom event creation (e.g. A & (!B | !C)). Note that LSB of this field affects bit 0 of the selected input bus.

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0b</td>
<td>Pass-through</td>
<td>Input bit is passed through to comparator as is</td>
</tr>
<tr>
<td>1b</td>
<td>Negated</td>
<td>Input bit is negated before passing to comparator</td>
</tr>
</tbody>
</table>

| 20:19 | **Source Select** |
|       | **Format:** U2 |

Selects the input signals to the Boolean event definition logic (see block diagram in the Custom Event Counters section).

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>01b</td>
<td>Prev Event</td>
<td>Selects the conditioned/flopped input from the previous CEC block as the input bus to this CEC block</td>
</tr>
<tr>
<td>11b</td>
<td>Reserved</td>
<td></td>
</tr>
</tbody>
</table>

| 18:3 | **Compare Value** |
|      | **Format:** U16 |

The value in this field is compared the 16-bit conditioned input bus that are fed into the comparator (see block diagram in the Custom Event Counters section). The type of comparison that is done is controlled by the Compare Function. When the compare function is true, then the signal for the custom event is asserted. This signal in turn can be counted by the B0 performance counter or fed into other CEC blocks.
## CEC1-0 - Customizable Event Creation 1-0

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>000b</td>
<td>Any Are Equal</td>
<td>Compare and assert output if any are equal (Can be used as OR function)</td>
</tr>
<tr>
<td>001b</td>
<td>Greater Than</td>
<td>Compare and assert output if greater than</td>
</tr>
<tr>
<td>010b</td>
<td>Equal</td>
<td>Compare and assert output if equal to (Can also be used as AND function)</td>
</tr>
<tr>
<td>011b</td>
<td>Greater Than or Equal</td>
<td>Compare and assert output if greater than or equal</td>
</tr>
<tr>
<td>100b</td>
<td>Less Than</td>
<td>Compare and assert output if less than</td>
</tr>
<tr>
<td>101b</td>
<td>Not Equal</td>
<td>Compare and assert output if not equal</td>
</tr>
<tr>
<td>110b</td>
<td>Less Than or Equal</td>
<td>Compare and assert output if less than or equal</td>
</tr>
<tr>
<td>111b</td>
<td>Reserved</td>
<td></td>
</tr>
</tbody>
</table>
Customizable Event Creation 2-0

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:21</td>
<td><strong>Negate</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: U11</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The 11-bit array defined in this field allows input bus bits [10:0] to be individually negated in order to facilitate more complicated custom event creation (e.g. (A &amp; (\neg B</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Value</td>
</tr>
<tr>
<td>0b</td>
<td>Pass-through</td>
<td>Input bit is passed through to comparator as is</td>
</tr>
<tr>
<td>1b</td>
<td>Negated</td>
<td>Input bit is negated before passing to comparator</td>
</tr>
<tr>
<td>20:19</td>
<td><strong>Source Select</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Format: U2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Selects the input signals to the Boolean event definition logic (see block diagram in the Custom Event Counters section).</td>
<td></td>
</tr>
<tr>
<td>Value</td>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>01b</td>
<td>Prev Event</td>
<td>Selects the conditioned/flopped input from the previous CEC block as the input bus to this CEC block</td>
</tr>
<tr>
<td>11b</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>18:3</td>
<td><strong>Compare Value</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Format: U16</td>
<td></td>
</tr>
<tr>
<td></td>
<td>The value in this field is compared the 16-bit conditioned input bus that are fed into the comparator (see block diagram in the Custom Event Counters section). The type of comparison that is done is controlled by the Compare Function. When the compare function is true, then the signal for the custom event is asserted. This signal in turn can be counted by the B0 performance counter or fed into other CEC blocks.</td>
<td></td>
</tr>
</tbody>
</table>
## CEC2-0 - Customizable Event Creation 2-0

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>000b</td>
<td>Any Are Equal</td>
<td>Compare and assert output if any are equal (Can be used as OR function)</td>
</tr>
<tr>
<td>001b</td>
<td>Greater Than</td>
<td>Compare and assert output if greater than</td>
</tr>
<tr>
<td>010b</td>
<td>Equal</td>
<td>Compare and assert output if equal to (Can also be used as AND function)</td>
</tr>
<tr>
<td>011b</td>
<td>Greater Than or Equal</td>
<td>Compare and assert output if greater than or equal</td>
</tr>
<tr>
<td>100b</td>
<td>Less Than</td>
<td>Compare and assert output if less than</td>
</tr>
<tr>
<td>101b</td>
<td>Not Equal</td>
<td>Compare and assert output if not equal</td>
</tr>
<tr>
<td>110b</td>
<td>Less Than or Equal</td>
<td>Compare and assert output if less than or equal</td>
</tr>
<tr>
<td>111b</td>
<td>Reserved</td>
<td></td>
</tr>
</tbody>
</table>
Customizable Event Creation 3-0

**CEC3-0 - Customizable Event Creation 3-0**

- **Register Space:** MMIO: 0/2/0
- **Project:** CHV, BSW
- **Source:** PRM
- **Default Value:** 0x00000000
- **Access:** R/W
- **Size (in bits):** 32
- **Address:** 02788h
- **Valid Projects:** CHV, BSW

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:21</td>
<td><strong>Negate</strong></td>
</tr>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
<td></td>
</tr>
<tr>
<td>Format:</td>
<td>U11</td>
<td></td>
</tr>
<tr>
<td>The 11-bit array defined in this field allows input bus bits [10:0] to be individually negated in order to facilitate more complicated custom event creation (e.g. A &amp; (!B</td>
<td>!C)). Note that LSB of this field affects bit 0 of the selected input bus.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0b</td>
<td>Pass-through</td>
<td>Input bit is passed through to comparator as is</td>
</tr>
<tr>
<td>1b</td>
<td>Negated</td>
<td>Input bit is negated before passing to comparator</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>20:19</th>
<th>Source Select</th>
</tr>
</thead>
<tbody>
<tr>
<td>Format:</td>
<td>U2</td>
</tr>
<tr>
<td>Selects the input signals to the Boolean event definition logic (see block diagram in the Custom Event Counters section).</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>01b</td>
<td>Prev Event</td>
<td>Selects the conditioned/flopped input from the previous CEC block as the input bus to this CEC block</td>
</tr>
<tr>
<td>11b</td>
<td>Reserved</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>18:3</th>
<th>Compare Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Format:</td>
<td>U16</td>
</tr>
<tr>
<td>The value in this field is compared the 16-bit conditioned input bus that are fed into the comparator (see block diagram in the Custom Event Counters section). The type of comparison that is done is controlled by the Compare Function. When the compare function is true, then the signal for the custom event is asserted. This signal in turn can be counted by the B0 performance counter or fed into other CEC blocks.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## CEC3-0 - Customizable Event Creation 3-0

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>000b</td>
<td>Any Are Equal</td>
<td>Compare and assert output if any are equal (Can be used as OR function)</td>
</tr>
<tr>
<td>001b</td>
<td>Greater Than</td>
<td>Compare and assert output if greater than</td>
</tr>
<tr>
<td>010b</td>
<td>Equal</td>
<td>Compare and assert output if equal to (Can also be used as AND function)</td>
</tr>
<tr>
<td>011b</td>
<td>Greater Than or Equal</td>
<td>Compare and assert output if greater than or equal</td>
</tr>
<tr>
<td>100b</td>
<td>Less Than</td>
<td>Compare and assert output if less than</td>
</tr>
<tr>
<td>101b</td>
<td>Not Equal</td>
<td>Compare and assert output if not equal</td>
</tr>
<tr>
<td>110b</td>
<td>Less Than or Equal</td>
<td>Compare and assert output if less than or equal</td>
</tr>
<tr>
<td>111b</td>
<td>Reserved</td>
<td></td>
</tr>
</tbody>
</table>
## Customizable Event Creation 4-0

### CEC4-0 - Customizable Event Creation 4-0

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>02790h</td>
</tr>
</tbody>
</table>

This register is used to define custom counter event 4, bit definitions in this register refer to the CEC block diagram in the Custom Event Counters section.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:21</td>
<td><strong>Negate</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format:</strong> U11</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The 11-bit array defined in this field allows input bus bits [10:0] to be individually negated in order to facilitate more complicated custom event creation (e.g. A &amp; (!B</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Value</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Name</strong></td>
</tr>
<tr>
<td></td>
<td>0b</td>
<td>Pass-through</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Input bit is passed through to comparator as is</td>
</tr>
<tr>
<td></td>
<td>1b</td>
<td>Negated</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Input bit is negated before passing to comparator</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>20:19</td>
<td><strong>Source Select</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format:</strong> U2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Selects the input signals to the Boolean event definition logic (see block diagram in the Custom Event Counters section).</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Value</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Name</strong></td>
</tr>
<tr>
<td></td>
<td>01b</td>
<td>Prev Event</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Selects the conditioned/flopped input from the last CEC block as the input bus to CEC0 block</td>
</tr>
<tr>
<td></td>
<td>11b</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>18:3</td>
<td><strong>Compare Value</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format:</strong> U16</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The value in this field is compared the 16-bit conditioned input bus that are fed into the comparator (see block diagram in the Custom Event Counters section). The type of comparison that is done is controlled by the Compare Function. When the compare function is true, then the signal for the custom event is asserted. This signal in turn can be counted by the B0 performance counter or fed into other CEC blocks.</td>
</tr>
</tbody>
</table>
## CEC4-0 - Customizable Event Creation 4-0

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>000b</td>
<td>Any Are Equal</td>
<td>Compare and assert output if any are equal (Can be used as OR function)</td>
</tr>
<tr>
<td>001b</td>
<td>Greater Than</td>
<td>Compare and assert output if greater than</td>
</tr>
<tr>
<td>010b</td>
<td>Equal</td>
<td>Compare and assert output if equal to (Can also be used as AND function)</td>
</tr>
<tr>
<td>011b</td>
<td>Greater Than or Equal</td>
<td>Compare and assert output if greater than or equal</td>
</tr>
<tr>
<td>100b</td>
<td>Less Than</td>
<td>Compare and assert output if less than</td>
</tr>
<tr>
<td>101b</td>
<td>Not Equal</td>
<td>Compare and assert output if not equal</td>
</tr>
<tr>
<td>110b</td>
<td>Less Than or Equal</td>
<td>Compare and assert output if less than or equal</td>
</tr>
<tr>
<td>111b</td>
<td>Reserved</td>
<td></td>
</tr>
</tbody>
</table>
# Customizable Event Creation 5-0

<table>
<thead>
<tr>
<th><strong>Register Space</strong></th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Project</strong></td>
<td>CHV, BSW</td>
</tr>
<tr>
<td><strong>Source</strong></td>
<td>PRM</td>
</tr>
<tr>
<td><strong>Default Value</strong></td>
<td>0x00000000</td>
</tr>
<tr>
<td><strong>Access</strong></td>
<td>R/W</td>
</tr>
<tr>
<td><strong>Size (in bits)</strong></td>
<td>32</td>
</tr>
</tbody>
</table>

This register is used to define custom counter event 5, bit definitions in this register refer to the CEC block diagram in the Custom Event Counters section.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td><strong>Negate</strong></td>
</tr>
<tr>
<td></td>
<td>31:21</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td></td>
<td>Format:</td>
<td>U11</td>
</tr>
</tbody>
</table>

The 11-bit array defined in this field allows input bus bits [10:0] to be individually negated in order to facilitate more complicated custom event creation (e.g. A & (!B | !C)). Note that LSB of this field affects bit 0 of the selected input bus.

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
<th>Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Pass-through</td>
<td>Input bit is passed through to comparator as is</td>
<td>All</td>
</tr>
<tr>
<td>1</td>
<td>Negated</td>
<td>Input bit is negated before passing to comparator</td>
<td>All</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>20:19</td>
<td><strong>Source Select</strong></td>
</tr>
<tr>
<td></td>
<td>Format:</td>
<td>U2</td>
</tr>
</tbody>
</table>

Selects the input signals to the Boolean event definition logic (see block diagram in the Custom Event Counters section).

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
<th>Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>Prev Event</td>
<td>Selects the conditioned/flopped input from the previous CEC block as the input bus to this CEC block</td>
<td>All</td>
</tr>
<tr>
<td>11</td>
<td>Reserved</td>
<td>All</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>18:3</td>
<td><strong>Compare Value</strong></td>
</tr>
<tr>
<td></td>
<td>Format:</td>
<td>U16</td>
</tr>
</tbody>
</table>

The value in this field is compared the 16-bit conditioned input bus that are fed into the comparator (see block diagram in the Custom Event Counters section). The type of comparison that is done is controlled by the Compare Function. When the compare function is true, then the signal for the custom event is asserted. This signal in turn can be counted by the B0 performance counter or fed into other CEC blocks.
## CEC5-0 - Customizable Event Creation 5-0

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>000b</td>
<td>Any Are Equal</td>
<td>Compare and assert output if any are equal (Can be used as OR function)</td>
</tr>
<tr>
<td>001b</td>
<td>Greater Than</td>
<td>Compare and assert output if greater than</td>
</tr>
<tr>
<td>010b</td>
<td>Equal</td>
<td>Compare and assert output if equal to (Can also be used as AND function)</td>
</tr>
<tr>
<td>011b</td>
<td>Greater Than or Equal</td>
<td>Compare and assert output if greater than or equal</td>
</tr>
<tr>
<td>100b</td>
<td>Less Than</td>
<td>Compare and assert output if less than</td>
</tr>
<tr>
<td>101b</td>
<td>Not Equal</td>
<td>Compare and assert output if not equal</td>
</tr>
<tr>
<td>110b</td>
<td>Less Than or Equal</td>
<td>Compare and assert output if less than or equal</td>
</tr>
<tr>
<td>111b</td>
<td>Reserved</td>
<td></td>
</tr>
</tbody>
</table>
Customizable Event Creation 6-0

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:21</td>
<td>Negate</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: U11</td>
</tr>
</tbody>
</table>

The 11-bit array defined in this field allows input bus bits [10:0] to be individually negated in order to facilitate more complicated custom event creation (e.g. A & (!B | !C)). Note that LSB of this field affects bit 0 of the selected input bus.

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
<th>Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>0b</td>
<td>Pass</td>
<td>Input bit is passed through to comparator as is</td>
<td>All</td>
</tr>
<tr>
<td>1b</td>
<td>Negated</td>
<td>Input bit is negated before passing to comparator</td>
<td>All</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>20:19</th>
<th>Source Select</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Format: U2</td>
</tr>
</tbody>
</table>

Selects the input signals to the Boolean event definition logic (see block diagram in the Custom Event Counters section).

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
<th>Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>01b</td>
<td>Prev</td>
<td>Selects the conditioned/flopped input from the previous CEC block as the input to this CEC block</td>
<td>All</td>
</tr>
<tr>
<td>11b</td>
<td>Reserved</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>18:3</th>
<th>Compare Value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Format: U16</td>
</tr>
</tbody>
</table>

The value in this field is compared the 16-bit conditioned input bus that are fed into the comparator (see block diagram in the Custom Event Counters section). The type of comparison that is done is controlled by the Compare Function. When the compare function is true, then the signal for the custom event is asserted. This signal in turn can be counted by the B0 performance counter or fed into other CEC blocks.
### CEC6-0 - Customizable Event Creation 6-0

#### Compare Function

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>000b</td>
<td>Any Are Equal</td>
<td>Compare and assert output if any are equal (Can be used as OR function)</td>
</tr>
<tr>
<td>001b</td>
<td>Greater Than</td>
<td>Compare and assert output if greater than</td>
</tr>
<tr>
<td>010b</td>
<td>Equal</td>
<td>Compare and assert output if equal to (Can also be used as AND function)</td>
</tr>
<tr>
<td>011b</td>
<td>Greater Than or Equal</td>
<td>Compare and assert output if greater than or equal</td>
</tr>
<tr>
<td>100b</td>
<td>Less Than</td>
<td>Compare and assert output if less than</td>
</tr>
<tr>
<td>101b</td>
<td>Not Equal</td>
<td>Compare and assert output if not equal</td>
</tr>
<tr>
<td>110b</td>
<td>Less Than or Equal</td>
<td>Compare and assert output if less than or equal</td>
</tr>
<tr>
<td>111b</td>
<td>Reserved</td>
<td></td>
</tr>
</tbody>
</table>
# Customizable Event Creation 7-0

## CEC7-0 - Customizable Event Creation 7-0

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>027A8h</td>
</tr>
</tbody>
</table>

This register is used to define custom counter event 7, bit definitions in this register refer to the CEC block diagram in the Custom Event Counters section.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>31:21</td>
<td>Negate</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: U11</td>
</tr>
</tbody>
</table>

The 11-bit array defined in this field allows input bus bits [10:0] to be individually negated in order to facilitate more complicated custom event creation (e.g. A & (\!B | \!C)). Note that LSB of this field affects bit 0 of the selected input bus.

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0b</td>
<td>Pass-through</td>
<td>Input bit is passed through to comparator as is</td>
</tr>
<tr>
<td>1b</td>
<td>Negated</td>
<td>Input bit is negated before passing to comparator</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>20:19</td>
<td></td>
<td>Source Select</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: U2</td>
</tr>
</tbody>
</table>

Selects the input signals to the Boolean event definition logic (see block diagram in the Custom Event Counters section).

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>01b</td>
<td>Prev Event</td>
<td>Selects the conditioned/flopped input from the previous CEC block as the input bus to this CEC block</td>
</tr>
<tr>
<td>11b</td>
<td>Reserved</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>18:3</td>
<td></td>
<td>Compare Value</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: U16</td>
</tr>
</tbody>
</table>

The value in this field is compared the 16-bit conditioned input bus that are fed into the comparator (see block diagram in the Custom Event Counters section). The type of comparison that is done is controlled by the Compare Function. When the compare function is true, then the signal for the custom event is asserted. This signal in turn can be counted by the B0 performance counter or fed into other CEC blocks.
## CEC7-0 - Customizable Event Creation 7-0

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>000b</td>
<td>Any Are Equal</td>
<td>Compare and assert output if any are equal (Can be used as OR function)</td>
</tr>
<tr>
<td>001b</td>
<td>Greater Than</td>
<td>Compare and assert output if greater than</td>
</tr>
<tr>
<td>010b</td>
<td>Equal</td>
<td>Compare and assert output if equal to (Can also be used as AND function)</td>
</tr>
<tr>
<td>011b</td>
<td>Greater Than or Equal</td>
<td>Compare and assert output if greater than or equal</td>
</tr>
<tr>
<td>100b</td>
<td>Less Than</td>
<td>Compare and assert output if less than</td>
</tr>
<tr>
<td>101b</td>
<td>Not Equal</td>
<td>Compare and assert output if not equal</td>
</tr>
<tr>
<td>110b</td>
<td>Less Than or Equal</td>
<td>Compare and assert output if less than or equal</td>
</tr>
<tr>
<td>111b</td>
<td>Reserved</td>
<td></td>
</tr>
</tbody>
</table>
## CVS TLB LRA 0

### CVS_TLB_LRA_0 - CVS TLB LRA 0

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>30:24</td>
<td>CVS LRA1 Max</td>
<td>Default Value: 1011111b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: CHV, BSW, GT2:B</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Maximum value of programmable LRA1.</td>
</tr>
<tr>
<td>23</td>
<td>Reserved</td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>22:16</td>
<td>CVS LRA1 Min</td>
<td>Default Value: 0100000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Minimum value of programmable LRA1.</td>
</tr>
<tr>
<td>15</td>
<td>Reserved</td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>14:8</td>
<td>CVS LRA0 Max</td>
<td>Default Value: 0011111b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Maximum value of programmable LRA0.</td>
</tr>
<tr>
<td>7</td>
<td>Reserved</td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>CVS_TLB_LRA_0 - CVS TLB LRA 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6:0 CVS LRA0 Min</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default Value:</td>
<td>0000000b</td>
<td></td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>Minimum value of programmable LRA0.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## CVS TLB LRA 1

### Register Information
- **Register Space**: MMIO: 0/2/0
- **Project**: CHV, BSW, GT2:B
- **Source**: PRM
- **Default Value**: 0x7F007F60
- **Size (in bits)**: 32
- **Address**: 04A24h

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31</td>
<td><strong>Reserved</strong>&lt;br&gt;Default Value: 0b&lt;br&gt;Access: RO</td>
</tr>
<tr>
<td>30:24</td>
<td>CVS LRA3 Max&lt;br&gt; Default Value: 1111111b&lt;br&gt;Access: R/W&lt;br&gt;For Future Use.</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td><strong>Reserved</strong>&lt;br&gt;Default Value: 0b&lt;br&gt;Access: RO</td>
<td></td>
</tr>
<tr>
<td>22:16</td>
<td>CVS LRA3 Min&lt;br&gt; Default Value: 0000000b&lt;br&gt;Access: R/W&lt;br&gt;For Future Use.</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td><strong>Reserved</strong>&lt;br&gt;Default Value: 0b&lt;br&gt;Access: RO</td>
<td></td>
</tr>
<tr>
<td>14:8</td>
<td>CVS LRA2 Max&lt;br&gt; Default Value: 1111111b&lt;br&gt;Access: R/W&lt;br&gt;Maximum value of programmable LRA2.</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td><strong>Reserved</strong>&lt;br&gt;Default Value: 0b&lt;br&gt;Access: RO</td>
<td></td>
</tr>
<tr>
<td>6:0</td>
<td><strong>CVS LRA2 Min</strong></td>
<td></td>
</tr>
<tr>
<td>-----</td>
<td>------------------</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Default Value:</strong></td>
<td>1100000b</td>
</tr>
<tr>
<td></td>
<td><strong>Access:</strong></td>
<td>R/W</td>
</tr>
</tbody>
</table>

Minimum value of programmable LRA2.
# CVS TLB LRA 2

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:8</td>
<td>Reserved</td>
<td>0000000h</td>
<td>RO</td>
</tr>
<tr>
<td>7:6</td>
<td></td>
<td>RS LRA</td>
<td>00b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Which LRA should RS use</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5:4</td>
<td></td>
<td>CS LRA</td>
<td>01b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Which LRA should CS use</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3:2</td>
<td></td>
<td>SOL LRA</td>
<td>10b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Which LRA should SOL use</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1:0</td>
<td></td>
<td>VF LRA</td>
<td>10b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Which LRA should VF use</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Register Space: MMIO: 0/2/0
Project: CHV, BSW, :GT2:B
Source: PRM
Default Value: 0x0000001A
Size (in bits): 32
Address: 04A28h
Depth/Early Depth TLB Partitioning Register

<table>
<thead>
<tr>
<th>ZSHR - Depth/Early Depth TLB Partitioning Register</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> RenderCS</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000020</td>
</tr>
<tr>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Trusted Type:</strong> 1</td>
</tr>
<tr>
<td><strong>Address:</strong> 04050h</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:6</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format:</strong> MBZ</td>
</tr>
<tr>
<td></td>
<td>5:0</td>
<td><strong>Number of TLB Entries Out of 64 used for Depth TLB</strong></td>
</tr>
<tr>
<td></td>
<td>Default Value: 32</td>
<td></td>
</tr>
</tbody>
</table>

The rest are be used for Early Depth/Stencil TLB. Default value is 32.
## DID - DID

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>DEVICEID UB</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 22h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DID_MSB: Identifier assigned to the dev2 PCI. Punit will update this bitfield before handing control to BIOS.</td>
</tr>
<tr>
<td>23:16</td>
<td></td>
<td><strong>DEVICEID LB</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: B0h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DID_LSB: Identifier assigned to the dev2 PCI. This bitfield is updated from metal straps. CHV, BSW allocated values: 0xB0, 0xB1, 0xB2, 0xB3.</td>
</tr>
<tr>
<td>23:16</td>
<td></td>
<td><strong>DEVICEID LB</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: B0h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DID_LSB: Identifier assigned to the dev2 PCI. Bits[7:2] is updated from metal straps. Bits[1:0] will be based on market segment SKU (ie via. fusing) CHV, BSW allocated values: 0xB0, 0xB1, 0xB2, 0xB3.</td>
</tr>
<tr>
<td>15:0</td>
<td>VENDORID</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>----------</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default Value: 8086h</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access: RO</td>
<td></td>
</tr>
</tbody>
</table>

VID: PCI standard identification for Intel
DS Invocation Counter

<table>
<thead>
<tr>
<th>DS_INVOCATION_COUNT - DS Invocation Counter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: RenderCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000, 0x00000000</td>
</tr>
<tr>
<td>Access: R/W</td>
</tr>
<tr>
<td>Size (in bits): 64</td>
</tr>
<tr>
<td>Trusted Type: 1</td>
</tr>
<tr>
<td>Address: 02308h</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
</table>
| 0     | 63:32| DS Invocation Count UDW  
Number of domain points shaded by the DS threads. Updated only when DS Function Enable and Statistics Enable are set in 3DSTATE_DS |
|       | 31:0 | DS Invocation Count LDW  
Number of domain points shaded by the DS threads. Updated only when DS Function Enable and Statistics Enable are set in 3DSTATE_DS |
## DX9 Constants Not Consumed By RCS

### DX9CONST_PRODUCE_COUNT - DX9 Constants Not Consumed By RCS

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>RenderCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Trusted Type:</td>
<td>1</td>
</tr>
<tr>
<td>Address:</td>
<td>02484h</td>
</tr>
</tbody>
</table>

This register keeps track of the outstanding DX9 Constants produced by RS which are not yet consumed by Render Command Streamer. This register is part of the render context save and restore. This register should not be programmed by SW.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
</table>
| 0     | 31:0| **DX9 Constants Produce Count**  
This register keeps track of the outstanding DX9 Constants produced by RS which are not yet consumed by Render Command Streamer. This register is part of the render context save and restore. |
## DX9 Constants Prsed By RCS

<table>
<thead>
<tr>
<th><strong>DX9CONST_PARSE_COUNT - DX9 Constants Prsed By RCS</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> RenderCS</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Trusted Type:</strong> 1</td>
</tr>
<tr>
<td><strong>Address:</strong> 02494h</td>
</tr>
</tbody>
</table>

This register keeps track of the DX9 Constant commands parsed by RCS prior to encountering the Draw Call in an RS enabled Batch Buffer which has DX9 Pool Alloc Valid. DX9 parse count should be less than equal to the DX9 produce count for Command Streamer to make progress on a 3D_PRIMITIVE command. DX9 Parse count is subtracted from the DX9 Produce Count upon parsing 3D_PRIMITIVE command. This register is part of the render context save and restore. This register should not be programmed by SW.

<table>
<thead>
<tr>
<th><strong>DWord</strong></th>
<th><strong>Bit</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>DX9 Constants Produce Count</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>This register keeps track of the DX9 Constant commands parsed by RCS prior to encountering the Draw Call in an RS enabled Batch Buffer which has DX9 Pool Alloc Valid. DX9 parse count should be less than equal to the DX9 produce count for Command Streamer to make progress on a 3D_PRIMITIVE command. DX9 Parse count is subtracted from the DX9 Produce Count upon parsing 3D_PRIMITIVE command.</td>
</tr>
</tbody>
</table>
ECO Bits - Bus Reset Domain with lock bit

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
</table>
| 0     | 31  | *ECO Bits - Bus Reset Domain - LOCK BIT*  
  Access: R/W Lock |
|       | 30  | **Reserved**  
  Access: RO |
|       | 29  | **Flush and block gfx pipes during cpd enter**  
  Access: R/W Lock  
  0 = GFX Pipes will not blocked during CPD enter  
  1 = GFX Pipes will be blocked and flushed during CPD enter. They will be unblocked again during CPD exit.  
  This bit must not be set to 1 |
|       | 28  | **Block gfx from accessing memory during cpd enter**  
  Access: R/W Lock  
  0 = GFX will not be blocked from accessing memory during CPD enter  
  1 = GFX will be blocked from accessing memory (go=0) during CPD enter. They will be unblocked again during CPD exit (go=1). |
|       | 27:26 | *ECO Bits - Bus Reset Domain1*  
  Access: R/W Lock  
  Register bits that have no connection to design. Used to enable/disable changes to the design that were put in during the ECO process.  
  pmcr_eco_busrst[26:25] |
|       | 25  | **CPD GAM GO Messaging Enable**  
  Access: R/W Lock  
  0 = No GO Messages to GAM during during CPD enter/exit flows (including slice shutdown) (default)  
  1 = GO Messages to GAM will occur during CPD enter/exit flows (including slice shutdown) |
## ECOBUS - ECO Bits - Bus Reset Domain with lock bit

<table>
<thead>
<tr>
<th>24:0</th>
<th><strong>ECO Bits - Bus Reset Domain</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Access: R/W Lock</td>
</tr>
</tbody>
</table>

### Description

Register bits that have no connection to design. Used to enable/disable changes to the design that were put in during the ECO process.

- **pmcr_eco_busrst[24:0]**

<table>
<thead>
<tr>
<th>Bit 21</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>VCS/CS/BCS Idleness, as well as CP/MBC/DT Idleness required for RC6 entry</td>
</tr>
<tr>
<td>0</td>
<td>VCS/CS/BCS Idleness required for RC6 entry</td>
</tr>
</tbody>
</table>

- **[20] - Reserved**
- **[19] - Reserved**
- **[18] - Reserved**
- **1** - Disable compensation
- **0** - Enable compensation of respective counters

- **[17] - Reserved**
- **1** -- Disable WIDI Context Save/Restore
- **0** -- Enable WIDI Context Save/Restore (Default)

- **[16:15]** -- Bits to indicate how many EU's, in each SubSlice, to bring up as part of Render Standby exit
- **00** -- Bring up 8 EU's (Default)
- **01** -- Bring up only 6 EU's
- **10** -- Bring up only 4 EU's
- **11** -- Bring up just 2 EU's

- **[14:0]** are spare bits for ECO process.
# ECO Bits - Device Reset Domain

<table>
<thead>
<tr>
<th>Description</th>
<th>DWord</th>
<th>Bit</th>
<th>Access</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ECODEV</strong></td>
<td>0</td>
<td>31:0</td>
<td></td>
<td>R/W</td>
<td>Register bits that have no connection to design. Used to enable/disable changes to the design that were put in during the ECO process. pmcr_eco_bits[31:0]</td>
</tr>
</tbody>
</table>

**Register Space:** MMIO: 0/2/0  
**Project:** CHV, BSW  
**Source:** PRM  
**Default Value:** 0x00000000  
**Size (in bits):** 32  
**Address:** 0A184h-0A187h
## ECO Message Register

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>16</td>
</tr>
<tr>
<td>Address:</td>
<td>08040h</td>
</tr>
</tbody>
</table>

Message registers have bit-wise masking applied for writes. The register consists of 16 bits of data in [15:0], and 16 bits of corresponding masks in [31:16]. In order to set bit0, for example, the data would be 0x0001_0001. In order to clear bit0, for example, the data would be 0x0001_0000. Note that mask bit is the data bit offset + 16. Message registers are protected from non-GT writes via the Message Channel.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>15</td>
<td><strong>Placeholder for ECO Bit 15</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ECO Bits 1'b0 : &lt;default&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Register definition will be modified if ECOs are required.</td>
</tr>
<tr>
<td>14</td>
<td>14</td>
<td><strong>Placeholder for ECO Bit 14</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ECO Bits 1'b0 : &lt;default&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Register definition will be modified if ECOs are required.</td>
</tr>
<tr>
<td>13</td>
<td>13</td>
<td><strong>Placeholder for ECO Bit 13</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ECO Bits 1'b0 : &lt;default&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Register definition will be modified if ECOs are required.</td>
</tr>
<tr>
<td>12</td>
<td>12</td>
<td><strong>Placeholder for ECO Bit 12</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ECO Bits 1'b0 : &lt;default&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Register definition will be modified if ECOs are required.</td>
</tr>
<tr>
<td>11</td>
<td>11</td>
<td><strong>Placeholder for ECO Bit 11</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ECO Bits 1'b0 : &lt;default&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Register definition will be modified if ECOs are required.</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td><strong>Placeholder for ECO Bit 10</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ECO Bits 1'b0 : &lt;default&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Register definition will be modified if ECOs are required.</td>
</tr>
<tr>
<td></td>
<td>ECO_MSG - ECO Message Register</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>--------------------------------</td>
<td>---</td>
</tr>
<tr>
<td>9</td>
<td><strong>Placeholder for ECO Bit 9</strong></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td>ECO Bits 1'b0 : &lt;default&gt;</td>
<td>Register definition will be modified if ECOs are required.</td>
</tr>
<tr>
<td>8</td>
<td><strong>Placeholder for ECO Bit 8</strong></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td>ECO Bits 1'b0 : &lt;default&gt;</td>
<td>Register definition will be modified if ECOs are required.</td>
</tr>
<tr>
<td>7</td>
<td><strong>Placeholder for ECO Bit 7</strong></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td>ECO Bits 1'b0 : &lt;default&gt;</td>
<td>Register definition will be modified if ECOs are required.</td>
</tr>
<tr>
<td>6</td>
<td><strong>Placeholder for ECO Bit 6</strong></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td>ECO Bits 1'b0 : &lt;default&gt;</td>
<td>Register definition will be modified if ECOs are required.</td>
</tr>
<tr>
<td>5</td>
<td><strong>Placeholder for ECO Bit 5</strong></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td>ECO Bits 1'b0 : &lt;default&gt;</td>
<td>Register definition will be modified if ECOs are required.</td>
</tr>
<tr>
<td>4</td>
<td><strong>Placeholder for ECO Bit 4</strong></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td>ECO Bits 1'b0 : &lt;default&gt;</td>
<td>Register definition will be modified if ECOs are required.</td>
</tr>
<tr>
<td>3</td>
<td><strong>Placeholder for ECO Bit 3</strong></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td>ECO Bits 1'b0 : &lt;default&gt;</td>
<td>Register definition will be modified if ECOs are required.</td>
</tr>
<tr>
<td>2</td>
<td><strong>Placeholder for ECO Bit 2</strong></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td>ECO Bits 1'b0 : &lt;default&gt;</td>
<td>Register definition will be modified if ECOs are required.</td>
</tr>
<tr>
<td>1</td>
<td><strong>Placeholder for ECO Bit 1</strong></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td>ECO Bits 1'b0 : &lt;default&gt;</td>
<td>Register definition will be modified if ECOs are required.</td>
</tr>
<tr>
<td>0</td>
<td><strong>Placeholder for ECO Bit 0</strong></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td>ECO Bits 1'b0 : &lt;default&gt;</td>
<td>Register definition will be modified if ECOs are required.</td>
</tr>
</tbody>
</table>
## ECO Reserved

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>09898h</td>
</tr>
</tbody>
</table>

### ECO Reserved bits

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Access:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>ECO Reserved Bits</td>
<td>R/WC</td>
<td></td>
</tr>
</tbody>
</table>

Access: R/WC
## ECREQ

### ECREQ - ECREQ

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>1300A0h</td>
</tr>
</tbody>
</table>

This register contains the energy counter request bit.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:1</td>
<td><strong>SPARE</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Spare bits for Duty cycle control feature.</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td><strong>Energy Count Request</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WO</td>
</tr>
</tbody>
</table>

WO access type of this register means that it will always return a zero when read.
A write will generate a one CZ clock wide pulse to indicate that Punit requests for Gunit to push the energy counters.
This pulse will only be created when both the data attempted to be written is one and when the byte enables allow the write.
## Element Descriptor Register

<table>
<thead>
<tr>
<th>Element Descriptor Register</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong></td>
<td>MMIO: 0/2/0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Project:</strong></td>
<td>CHV, BSW</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Source:</strong></td>
<td>PRM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Default Value:</strong></td>
<td>0x00000000, 0x00000000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Access:</strong></td>
<td>RO</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Size (in bits):</strong></td>
<td>64</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Address:** 04500h  
**Name:** BCS Element Descriptor Register  
**ShortName:** BCS_ELEM_DESCRIPTOR

**Address:** 04400h  
**Name:** RCS Element Descriptor Register  
**ShortName:** RCS_ELEM_DESCRIPTOR

**Address:** 04440h  
**Name:** VCS Element Descriptor Register  
**ShortName:** VCS_ELEM_DESCRIPTOR

**Address:** 044C0h  
**Name:** VECS Element Descriptor Register  
**ShortName:** VECS_ELEM_DESCRIPTOR

Element Information: The register is populated by command streamer and consumed by GAM

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
</table>
| 0     | 63:32 | **Context ID**  
Context identification number assigned to separate this context from others. Context IDs need to be recycled in such a way that there could not be two active context with the same ID. This is a unique identification number by which a context is identified and referenced  
| 31:12 |      | **LRCA**  
Command Streamer Only  
| 11:9  |      | **Function Number**  
GFX device is considered to be on Bus0 with device number of 2. Function number is normally assigned as "0" however for gfx virtualization; there would be different function numbers which needs to be attached to context. Not used in Gen8. |
### ELEM_DESCRIPTOR - Element Descriptor Register

#### Privileged Context / GGTT vs PPGTT mode
In Legacy Context: Defines the page tables to be used. This is how page walker come to know PPGTT vs GGTT selection for the entire context. In Advanced Context: Defines the privilege level for the context.

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>[Default]</td>
<td>Use Global GTT (In Legacy Context) User Mode Context (In Advanced Context)</td>
</tr>
<tr>
<td>1h</td>
<td></td>
<td>Use Per-Process GTT (In Legacy Context) Supervisor Mode Context (In Advanced Context)</td>
</tr>
</tbody>
</table>

#### Fault Model
In Legacy Context: Defines the page tables to be used. In Advanced Context: Defines the privilege level for the context.

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>00h</td>
<td>[Default]</td>
<td>Fault and Hang. Same mode as gen7.5</td>
</tr>
<tr>
<td>01h</td>
<td></td>
<td>Fault and Halt/Wait. Same mode as gen7.5</td>
</tr>
<tr>
<td>10h</td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td>11h</td>
<td></td>
<td>Fault and Continue (reserved for gen8) - does not generate a page request to IOMMU.</td>
</tr>
</tbody>
</table>

#### Deeper IA coherency Support
In Advanced Context: Defines the level of IA coherency.

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>[Default]</td>
<td>IA coherency is provided at LLC level for all streams of GPU (i.e. gen7.5 like mode)</td>
</tr>
<tr>
<td>1h</td>
<td></td>
<td>IA coherency is provided at L3 level for EU data accesses of GPU</td>
</tr>
</tbody>
</table>

#### A and D Support / 32 and 64b Address Support
In Legacy Context: Defines 32b vs 64b (48b canonical) addressing format. In Advanced Context: Defines A/D bit support.

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>[Default]</td>
<td>32b addressing format (In Legacy Context) A/D bit management in page tables is NOT supported (In Advanced Context)</td>
</tr>
<tr>
<td>1h</td>
<td></td>
<td>64b (48b canonical) addressing format (In Legacy Context) A/D bit management in page tables is supported (In Advanced Context)</td>
</tr>
</tbody>
</table>

#### Context Type: Legacy vs Advanced
Defines the context type. Note that: Bits [8:4] differs in functions when legacy vs advanced context modes are selected.

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>[Default]</td>
<td>Advanced Context: Defines the rest of the advanced capabilities (i.e. OS page table support, fault models...). Note that advanced context is not bounded to GPGPU.</td>
</tr>
<tr>
<td>1h</td>
<td></td>
<td>Legacy Context: Defines the context as legacy mode which is similar to prior generations of gen8.</td>
</tr>
</tbody>
</table>
## ELEM_DESCRIPTOR - Element Descriptor Register

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>[Default]</td>
<td>Indicates exclist mode of scheduling.</td>
</tr>
<tr>
<td>1h</td>
<td></td>
<td>Indicates Ring Buffer mode of scheduling.</td>
</tr>
</tbody>
</table>

### Scheduling Mode

<table>
<thead>
<tr>
<th>Project:</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Valid</td>
<td>Indicates that element descriptor is valid. If GAM is programmed with an invalid descriptor, it will continue but flag an error.</td>
</tr>
</tbody>
</table>
Error_Identity_Reg

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>020B0h-020B3h</td>
</tr>
<tr>
<td>Name:</td>
<td>Error Identity Register</td>
</tr>
<tr>
<td>ShortName:</td>
<td>EIR_RCSUNIT</td>
</tr>
<tr>
<td>Address:</td>
<td>120B0h-120B3h</td>
</tr>
<tr>
<td>Name:</td>
<td>Error Identity Register</td>
</tr>
<tr>
<td>ShortName:</td>
<td>EIR_VCSUNIT0</td>
</tr>
<tr>
<td>Address:</td>
<td>1A0B0h-1A0B3h</td>
</tr>
<tr>
<td>Name:</td>
<td>Error Identity Register</td>
</tr>
<tr>
<td>ShortName:</td>
<td>EIR_VECSPUNIT</td>
</tr>
<tr>
<td>Address:</td>
<td>1C0B0h-1C0B3h</td>
</tr>
<tr>
<td>Name:</td>
<td>Error Identity Register</td>
</tr>
<tr>
<td>ShortName:</td>
<td>EIR_VCSUNIT1</td>
</tr>
<tr>
<td>Address:</td>
<td>220B0h-220B3h</td>
</tr>
<tr>
<td>Name:</td>
<td>Error Identity Register</td>
</tr>
<tr>
<td>ShortName:</td>
<td>EIR_BCSUNIT</td>
</tr>
<tr>
<td>Address:</td>
<td>1820B0h</td>
</tr>
</tbody>
</table>

Error Identity. This register contains the persistent values of ESR error status bits that are unmasked via the EMR register. The logical OR of all (defined) bits in this register is reported in the Master Error bit of the Interrupt Status Register (ISR).

In order to clear an error condition, software must first clear the error by writing a 1 to the appropriate bit(s) in this field. If required, software should then proceed to clear the Master Error bit of the IIR.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:7</td>
<td>RESERVED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>Gunit_TLB_DataE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
</tbody>
</table>
## EIR - Error_Identity_Reg

| Access: R/W One Clear |  
|---|---|
| **Set if Gunit TLB has a data valid error** |  

### 5 Gunit_TLB_PTE

| Default Value: 0b |  
|---|---|
| **Access: R/W One Clear** |  
| **Set if Gunit TLB has a PTE translation error.** |  

### 4 PAGE_TABLE_ERROR

| Default Value: 0b |  
|---|---|
| **Access: R/W One Clear** |  
| **PTE: This bit is set when a Graphics Memory Mapping Error is detected and it’s not EMR masked.** |  
| **The cause of the error is indicated (to some extent) in the PGTBL_ER register.** |  
| **This error condition cannot be cleared except by reset (i.e., it is a fatal error)** |  
| **mir_EIR[4] = (((dsp_gvd_invldgtppte_int_dczfwoh and ~mir_EMR[4]) | mir_EIR[4]) and ~(sys_wdata[4] and mir_write and mirb0_decode and ~sys_data_mask[0]));** |  

### 3:1 RESERVED

| Default Value: 0h |  
|---|---|
| **Access: RO** |  
| **Reserved** |  

### 0 CLAIM_ERROR

| Default Value: 0b |  
|---|---|
| **Access: R/W One Clear** |  
| **If EMR[0]=1, this bit is set when an address within Gunit address space is accessed, but no memory mapped register exists in this address.** |  
| **This error condition can be cleared by writing 1b to this bit.** |  
| **Mir_EIR[0] = (((noRMclaim_err and ~mir_EMR[0]) | mir_EIR[0]) and ~(sys_wdata[0] and mir_write and mirb0_decode and ~sys_data_mask[0]));** |
# Error_Mask_Reg

## EMR - Error_Mask_Reg

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000071</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
</tbody>
</table>

| Address:       | 020B4h-020B7h |
| Name:          | Error Mask Register |
| ShortName:     | EMR_RCSUNIT   |

| Address:       | 120B4h-120B7h |
| Name:          | Error Mask Register |
| ShortName:     | EMR_VCSUNIT0   |

| Address:       | 1A0B4h-1A0B7h |
| Name:          | Error Mask Register |
| ShortName:     | EMR_VECSEUNIT  |

| Address:       | 1C0B4h-1C0B7h |
| Name:          | Error Mask Register |
| ShortName:     | EMR_VCSUNIT1  |

| Address:       | 220B4h-220B7h |
| Name:          | Error Mask Register |
| ShortName:     | EMR_BCSUNIT   |

| Address:       | 1820B4h       |

Error Mask. This register contains a bit mask that selects which error condition bits (from the ESR) are reported in the EIR.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:7</td>
<td><strong>RESERVED</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
</tbody>
</table>

| 6     | 6    | **Gunit_TLB_DataE**         |
|       |      | Default Value: 1b            |
|       |      | Access: R/W                  |

1 = Masked. This error will neither be reported in the EIR, nor set the master error condition in the IIR.
0 = Unmasked. This interrupt will be reported in the EIR and will set the master error condition in the IIR.
## EMR - Error_Mask_Reg

<table>
<thead>
<tr>
<th>Bit</th>
<th>Register</th>
<th>Default Value</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Gunit_TLB_PTE</td>
<td>1b</td>
<td>R/W</td>
<td>1 = Masked. This error will neither be reported in the EIR, nor set the master error condition in the IIR.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0 = Unmasked. This interrupt will be reported in the EIR and will set the master error condition in the IIR.</td>
</tr>
<tr>
<td>4</td>
<td>PAGE_TABLE_ERROR</td>
<td>1b</td>
<td>R/W</td>
<td>1 = Masked. This error will neither be reported in the EIR, nor set the master error condition in the IIR.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0 = Unmasked. This interrupt will be reported in the EIR and will set the master error condition in the IIR.</td>
</tr>
<tr>
<td>3:1</td>
<td>RESERVED</td>
<td>0h</td>
<td>RO</td>
<td>Reserved</td>
</tr>
<tr>
<td>0</td>
<td>CLAIM_ERROR</td>
<td>1b</td>
<td>R/W</td>
<td>1 = Masked. This error will neither be reported in the EIR, nor set the master error condition in the IIR.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0 = Unmasked. This interrupt will be reported in the EIR and will set the master error condition in the IIR.</td>
</tr>
</tbody>
</table>
## Error_Status_Reg

### ESR - Error_Status_Reg

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>020B8h-020BBh</td>
</tr>
<tr>
<td>Name:</td>
<td>Error Status Register</td>
</tr>
<tr>
<td>ShortName:</td>
<td>ESR_RCSUNIT</td>
</tr>
<tr>
<td>Address:</td>
<td>120B8h-120BBh</td>
</tr>
<tr>
<td>Name:</td>
<td>Error Status Register</td>
</tr>
<tr>
<td>ShortName:</td>
<td>ESR_VCSUNIT0</td>
</tr>
<tr>
<td>Address:</td>
<td>1A0B8h-1A0BBh</td>
</tr>
<tr>
<td>Name:</td>
<td>Error Status Register</td>
</tr>
<tr>
<td>ShortName:</td>
<td>ESR_VECSUNIT</td>
</tr>
<tr>
<td>Address:</td>
<td>1C0B8h-1C0BBh</td>
</tr>
<tr>
<td>Name:</td>
<td>Error Status Register</td>
</tr>
<tr>
<td>ShortName:</td>
<td>ESR_VCSUNIT1</td>
</tr>
<tr>
<td>Address:</td>
<td>220B8h-220BBh</td>
</tr>
<tr>
<td>Name:</td>
<td>Error Status Register</td>
</tr>
<tr>
<td>ShortName:</td>
<td>ESR_BCSUNIT</td>
</tr>
<tr>
<td>Address:</td>
<td>1820B8h</td>
</tr>
</tbody>
</table>

Error Status. This register contains the non-persistent values of all hardware-detected error condition bits.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:7</td>
<td>RESERVED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>Gunit_TLB_DataE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 = This error condition currently exists.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 = This error condition currently does not exist</td>
</tr>
<tr>
<td>Register</td>
<td>Default Value</td>
<td>Access</td>
</tr>
<tr>
<td>--------------------------</td>
<td>---------------</td>
<td>--------</td>
</tr>
<tr>
<td>Gunit_TLB_PTE</td>
<td>0b</td>
<td>RO</td>
</tr>
<tr>
<td>PAGE_TABLE_ERROR</td>
<td>0b</td>
<td>RO</td>
</tr>
<tr>
<td>RESERVED</td>
<td>0h</td>
<td>RO</td>
</tr>
<tr>
<td>CLAIM_ERROR</td>
<td>0b</td>
<td>RO</td>
</tr>
</tbody>
</table>
Error Identity Register

EIR - Error Identity Register

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>RenderCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W, RO</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>020B0h</td>
</tr>
</tbody>
</table>

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

### Restriction

Restriction EIR register contents are not power or render context save/restored. EIR register contents of an engine will get lost when the corresponding graphics engine (Render, Video, Video Enhancement, Blitter) is power down.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16 Reserved</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Format: MBZ</td>
<td></td>
</tr>
<tr>
<td>15:0</td>
<td>Error Identity Bits</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Format: Array of Error condition bits See the table titled Hardware-Detected Error Bits.</td>
<td></td>
</tr>
</tbody>
</table>

This register contains the persistent values of ESR error status bits that are unmasked via the EMR register. (See Table 3. Hardware-Detected Error Bits). The logical OR of all (defined) bits in this register is reported in the Master Error bit of the Interrupt Status Register. In order to clear an error condition, software must first clear the error by writing a 1 to the appropriate bit(s) in this field. If required, software should then proceed to clear the Master Error bit of the IIR. Reserved bits are RO.

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1h</td>
<td>Error occurred</td>
</tr>
</tbody>
</table>

### Programming Notes

Writing a 1 to a set bit will cause that error condition to be cleared. However, neither the Page Table Error bit (Bit 4) nor the Instruction Error bit (Bit 0) can be cleared except by reset (i.e., it is a fatal error).
Error Mask Register

<table>
<thead>
<tr>
<th>EMR - Error Mask Register</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space:</td>
</tr>
<tr>
<td>Project:</td>
</tr>
<tr>
<td>Source:</td>
</tr>
<tr>
<td>Default Value:</td>
</tr>
<tr>
<td>Access:</td>
</tr>
<tr>
<td>Size (in bits):</td>
</tr>
<tr>
<td>Address:</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:8</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: FFFFFFFh</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: Must Be One</td>
</tr>
</tbody>
</table>

**Programming Notes**

These bits are not implemented in HW and must be set to ‘1’

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7:0</td>
<td></td>
<td>Error Mask Bits</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: Array of error condition mask bits See the table titled Hardware-Detected Error Bits.</td>
</tr>
</tbody>
</table>

This register contains a bit mask that selects which error condition bits (from the ESR) are reported in the EIR.

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FFh</td>
<td>[Default]</td>
<td></td>
</tr>
<tr>
<td>0h</td>
<td>Not Masked</td>
<td>Will be reported in the EIR</td>
</tr>
<tr>
<td>1h</td>
<td>Masked</td>
<td>Will not be reported in the EIR</td>
</tr>
</tbody>
</table>
# Error Reporting Register

## ERR - Error Reporting Register

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:5</td>
<td><strong>Reserved</strong>&lt;br&gt;Access: RO&lt;br&gt;Reserved.</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td><strong>First Content Buffer Ready 0</strong>&lt;br&gt;Access: R/W&lt;br&gt;First Content Buffer Ready 0 (FRSNTBFR0).&lt;br&gt;First Content Buffer Ready: This bit gets set by the HW when the buffer is completely filled up and cleared by the driver when the contents of this buffer are copied out of memory.&lt;br&gt;Is set by lpfc_lpconf_buffer0_ready (pulse).&lt;br&gt;lpconf_lpfc_buffer0_ready (static signal to lpfc).</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td><strong>Second Buffer ready slice 0</strong>&lt;br&gt;Access: R/W&lt;br&gt;Second Content Buffer Ready slice 0 (SCNBFR0).&lt;br&gt;Second Content Buffer Ready: This bit gets set by the HW when the buffer is completely filled up and cleared by the driver when the contents of this buffer are copied out of memory.&lt;br&gt;Is set by lpfc_lpconf_buffer1_ready (pulse).&lt;br&gt;lpconf_lpfc_buffer1_ready (static signal to lpfc).</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td><strong>Write Expire Error Slice 0</strong>&lt;br&gt;Access: R/W&lt;br&gt;Write Expired Error slice 0 (WEERR0).&lt;br&gt;Write Expired Error: If DMA controller could not get a chance to push the write of 64Bytes to LTISEQ and data gets clobbered with the new expiration of the save timer, this error bit is set to indicate something went wrong.&lt;br&gt;Signal -lpfc_lpconf_wrexp_error.</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td><strong>Buffer full Error Slice 0</strong>&lt;br&gt;Access: R/W&lt;br&gt;Buffer full Error Slice 0 (BFFLERR0).&lt;br&gt;Set by lpfc_lpconf_error_buffer_full.&lt;br&gt;When all buffers are full lpfc sets this bit or if only 1 buffer is enabled then lpfc sets this bit when the buffer is full.</td>
</tr>
</tbody>
</table>
### ERR - Error Reporting Register

<table>
<thead>
<tr>
<th></th>
<th>Reserved</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access:</td>
</tr>
<tr>
<td></td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td>Reserved.</td>
</tr>
</tbody>
</table>

Access: RO

Reserved.
## Error Status Register

<table>
<thead>
<tr>
<th>ESR - Error Status Register</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: RenderCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Access: RO</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 020B8h</td>
</tr>
</tbody>
</table>

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

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

<table>
<thead>
<tr>
<th>TD_PM_MODE_EUCOUNT - EU Mask Programming</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Access: WO</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 0E4F8h</td>
</tr>
<tr>
<td>Name: EU Mask Programming Slice 0</td>
</tr>
<tr>
<td>ShortName: TD_PM_MODE_EUCOUNT_S0</td>
</tr>
<tr>
<td>Valid Projects: CHV, BSW</td>
</tr>
<tr>
<td>Address: 0E5F8h</td>
</tr>
<tr>
<td>Name: EU Mask Programming Slice 1</td>
</tr>
<tr>
<td>ShortName: TD_PM_MODE_EUCOUNT_S1</td>
</tr>
<tr>
<td>Valid Projects: CHV, BSW</td>
</tr>
<tr>
<td>Address: 0E6F8h</td>
</tr>
<tr>
<td>Name: EU Mask Programming Slice 2</td>
</tr>
<tr>
<td>ShortName: TD_PM_MODE_EUCOUNT_S2</td>
</tr>
<tr>
<td>Valid Projects: CHV, BSW</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
<tr>
<td>23</td>
<td></td>
<td>SubSlice 2 EU 7 Enable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: Enable</td>
</tr>
<tr>
<td>22</td>
<td></td>
<td>SubSlice 2 EU 6 Enable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: Enable</td>
</tr>
<tr>
<td>21</td>
<td></td>
<td>SubSlice 2 EU 5 Enable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: Enable</td>
</tr>
<tr>
<td>20</td>
<td></td>
<td>SubSlice 2 EU 4 Enable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: Enable</td>
</tr>
<tr>
<td>19</td>
<td></td>
<td>SubSlice 2 EU 3 Enable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: Enable</td>
</tr>
<tr>
<td>18</td>
<td></td>
<td>SubSlice 2 EU 2 Enable</td>
</tr>
</tbody>
</table>
## TD_PM_MODE_EUCOUNT - EU Mask Programming

<table>
<thead>
<tr>
<th>Address</th>
<th>Description</th>
<th>Format</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>SubSlice 2 EU 1 Enable</td>
<td>Enable</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>SubSlice 2 EU 0 Enable</td>
<td>Enable</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>SubSlice 1 EU 7 Enable</td>
<td>Enable</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>SubSlice 1 EU 6 Enable</td>
<td>Enable</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>SubSlice 1 EU 5 Enable</td>
<td>Enable</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>SubSlice 1 EU 4 Enable</td>
<td>Enable</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>SubSlice 1 EU 3 Enable</td>
<td>Enable</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>SubSlice 1 EU 2 Enable</td>
<td>Enable</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>SubSlice 1 EU 1 Enable</td>
<td>Enable</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>SubSlice 1 EU 0 Enable</td>
<td>Enable</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>SubSlice 0 EU 7 Enable</td>
<td>Enable</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>SubSlice 0 EU 6 Enable</td>
<td>Enable</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>SubSlice 0 EU 5 Enable</td>
<td>Enable</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>SubSlice 0 EU 4 Enable</td>
<td>Enable</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>SubSlice 0 EU 3 Enable</td>
<td>Enable</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>SubSlice 0 EU 2 Enable</td>
<td>Enable</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>SubSlice 0 EU 1 Enable</td>
<td>Enable</td>
<td></td>
</tr>
</tbody>
</table>
## TD_PM_MODE_EUCOUNT - EU Mask Programming

<table>
<thead>
<tr>
<th>0</th>
<th>SubSlice 0 EU 0 Enable</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Format: Enable</td>
</tr>
</tbody>
</table>

SubSlice 0 EU 0 Enable
## EU Metrics for Event0 High

### EUMETRICSEVENT0H - EU Metrics for Event0 High

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>138134h</td>
</tr>
</tbody>
</table>

Upper 32 bits of the EU Metrics Event0

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>EUMetrics, Event0 High</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Contains Upper 32 bits of EU Metrics, Event0</td>
</tr>
</tbody>
</table>
EU Metrics for Event0 Low

<table>
<thead>
<tr>
<th>EUMETRICSEVENT0L - EU Metrics for Event0 Low</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 138130h</td>
</tr>
</tbody>
</table>

Lower 14 bits of the EU Metrics Event0

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:18</td>
<td><strong>EUMetrics, Event0 Low</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Contains lower 14 bits of EU Metrics, Event0</td>
</tr>
<tr>
<td>17:0</td>
<td>Reserved</td>
<td>Default Value: 00000h</td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td>Reserved</td>
<td></td>
</tr>
</tbody>
</table>
## EU Metrics for Event1 High

<table>
<thead>
<tr>
<th><strong>EUMETRICSEVENT1H</strong> - EU Metrics for Event1 High</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 13813Ch</td>
</tr>
</tbody>
</table>

Upper 32 bits of the EU Metrics Event1

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>EUMetrics, Event1 High</strong></td>
</tr>
</tbody>
</table>

| Default Value: | 00000000h |
| Access:        | RO        |

Contains Upper 32 bits of EU Metrics, Event1
## EU Metrics for Event1 Low

<table>
<thead>
<tr>
<th>EUMETRICSEVENT1L - EU Metrics for Event1 Low</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 138138h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td><strong>EUMetrics, Event1 Low</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 0000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Contains lower 14 bits of EU Metrics, Event1</td>
</tr>
<tr>
<td></td>
<td>17:0</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 00000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
</tbody>
</table>
## EU Metrics for Event2 High

### EUMETRICSEVENT2H - EU Metrics for Event2 High

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>138144h</td>
</tr>
</tbody>
</table>

Upper 32 bits of the EU Metrics Event2

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>EUMetrics, Event2 High</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Contains Upper 32 bits of EU Metrics, Event2</td>
</tr>
</tbody>
</table>
EU Metrics for Event2 Low

<table>
<thead>
<tr>
<th>EUMETRICSEVENT2L - EU Metrics for Event2 Low</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 138140h</td>
</tr>
</tbody>
</table>

Lower 14 bits of the EU Metrics Event2

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:18</td>
<td><strong>EUMetrics, Event2 Low</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Contains lower 14 bits of EU Metrics, Event2</td>
</tr>
<tr>
<td>17:0</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
</tbody>
</table>
EU Metrics for Event3 High

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>13814Ch</td>
</tr>
</tbody>
</table>

Upper 32 bits of the EU Metrics Event3

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>EU Metrics, Event3 High</td>
<td>00000000h</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Contains Upper 32 bits of EU Metrics, Event3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
EU Metrics for Event3 Low

<table>
<thead>
<tr>
<th>EUMETRICSEVENT3L - EU Metrics for Event3 Low</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 138148h</td>
</tr>
</tbody>
</table>

Lower 14 bits of the EU Metrics Event3

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:18</td>
<td><strong>EUMetrics, Event3 Low</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Contains lower 14 bits of EU Metrics, Event3</td>
</tr>
<tr>
<td>17:0</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
</tbody>
</table>
## EU Metrics for Event4 High

### EUMETRICSEVENT4H - EU Metrics for Event4 High

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>138154h</td>
</tr>
</tbody>
</table>

Upper 32 bits of the EU Metrics Event4

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>EUMetrics, Event4 High</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Contains Upper 32 bits of EU Metrics, Event4</td>
</tr>
</tbody>
</table>
## EU Metrics for Event4 Low

<table>
<thead>
<tr>
<th>Description</th>
<th>DWord</th>
<th>Bit</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>EUMetrics, Event4 Low</td>
<td>0</td>
<td>31:18</td>
<td>0000h</td>
<td>RO</td>
</tr>
<tr>
<td>Reserved</td>
<td>17:0</td>
<td></td>
<td>000000h</td>
<td>RO</td>
</tr>
</tbody>
</table>

**Register Space:** MMIO: 0/2/0  
**Project:** CHV, BSW  
**Source:** PRM  
**Default Value:** 0x00000000  
**Size (in bits):** 32  
**Address:** 138150h  

Lower 14 bits of the EU Metrics Event4
EU Metrics for Event5 High

**EUMETRICSEVENT5H - EU Metrics for Event5 High**

Register Space: MMIO: 0/2/0  
Project: CHV, BSW  
Source: PRM  
Default Value: 0x00000000  
Size (in bits): 32  
Address: 13815Ch

Upper 32 bits of the EU Metrics Event5

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>EUMetrics, Event5 High</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Contains Upper 32 bits of EU Metrics, Event5</td>
</tr>
</tbody>
</table>
### EU Metrics for Event5 Low

#### EUMETRICSEVENT5L - EU Metrics for Event5 Low

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>138158h</td>
</tr>
</tbody>
</table>

Lower 14 bits of the EU Metrics Event5

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:18</td>
<td><strong>EUMetrics, Event5 Low</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Contains lower 14 bits of EU Metrics, Event5</td>
</tr>
<tr>
<td>17:0</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
</tbody>
</table>
## EU Metrics for Event6 High

<table>
<thead>
<tr>
<th><strong>EUMETRICSEVENT6H - EU Metrics for Event6 High</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 138164h</td>
</tr>
</tbody>
</table>

### Upper 32 bits of the EU Metrics Event6

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>EUMetrics, Event6 High</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Contains Upper 32 bits of EU Metrics, Event6</td>
</tr>
</tbody>
</table>
## EU Metrics for Event6 Low

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>138160h</td>
</tr>
</tbody>
</table>

### EU METRICSEVENT6L - EU Metrics for Event6 Low

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:18</td>
<td><strong>EUMetrics, Event6 Low</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 0000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Contains lower 14 bits of EU Metrics, Event6</td>
</tr>
<tr>
<td></td>
<td>17:0</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 00000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
</tbody>
</table>
## EU Metrics for Event7 High

<table>
<thead>
<tr>
<th><strong>EUMETRICSEVENT7H - EU Metrics for Event7 High</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 13816Ch</td>
</tr>
</tbody>
</table>

Upper 32 bits of the EU Metrics Event7

<table>
<thead>
<tr>
<th><strong>DWord</strong></th>
<th><strong>Bit</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>EUMetrics, Event7 High</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Contains Upper 32 bits of EU Metrics, Event7</td>
</tr>
</tbody>
</table>
**EU Metrics for Event7 Low**

<table>
<thead>
<tr>
<th><strong>EUMETRICSEVENT7L - EU Metrics for Event7 Low</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 138168h</td>
</tr>
</tbody>
</table>

Lower 14 bits of the EU Metrics Event7

<table>
<thead>
<tr>
<th><strong>DWord</strong></th>
<th><strong>Bit</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:18</td>
<td><strong>EUMetrics, Event7 Low</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 0000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Contains lower 14 bits of EU Metrics, Event7</td>
</tr>
<tr>
<td>17:0</td>
<td>Reserved</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 00000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
</tbody>
</table>
## EU Metrics for Event8 High

**EUMETRICSEVENT8H - EU Metrics for Event8 High**

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>EUMetrics, Event8 High</td>
</tr>
</tbody>
</table>

- **Default Value:** 00000000h
- **Access:** RO
- Contains Upper 32 bits of EU Metrics, Event8

**Register Space:** MMIO: 0/2/0  
**Project:** CHV, BSW  
**Source:** PRM  
**Default Value:** 0x00000000  
**Size (in bits):** 32  
**Address:** 138174h
# EU Metrics for Event8 Low

<table>
<thead>
<tr>
<th>EUMETRICSEVENT8L - EU Metrics for Event8 Low</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong></td>
</tr>
<tr>
<td><strong>Project:</strong></td>
</tr>
<tr>
<td><strong>Source:</strong></td>
</tr>
<tr>
<td><strong>Default Value:</strong></td>
</tr>
<tr>
<td><strong>Size (in bits):</strong></td>
</tr>
<tr>
<td><strong>Address:</strong></td>
</tr>
</tbody>
</table>

## Lower 14 bits of the EU Metrics Event8

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:18</td>
<td><strong>EUMetrics, Event8 Low</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 0000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Contains lower 14 bits of EU Metrics, Event8</td>
</tr>
<tr>
<td>17:0</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 00000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
</tbody>
</table>
# EU Metrics for Event9 High

<table>
<thead>
<tr>
<th>EUMETRICSEVENT9H - EU Metrics for Event9 High</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 13817Ch</td>
</tr>
</tbody>
</table>

Upper 32 bits of the EU Metrics Event9

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>EUMetrics, Event9 High</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Default Value:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>00000000h</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Access:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RO</td>
</tr>
</tbody>
</table>

Contains Upper 32 bits of EU Metrics, Event9
## EU Metrics for Event9 Low

### EUMETRICSEVENT9L - EU Metrics for Event9 Low

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>138178h</td>
</tr>
</tbody>
</table>

**Lower 14 bits of the EU Metrics Event9**

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:18</td>
<td><strong>EUMetrics, Event9 Low</strong></td>
<td>0000h</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Contains lower 14 bits of EU Metrics, Event9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17:0</td>
<td></td>
<td><strong>Reserved</strong></td>
<td>00000h</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## EU Metrics for Event10 High

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>138184h</td>
</tr>
</tbody>
</table>

**Upper 32 bits of the EU Metrics Event10**

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>EUMetrics, Event10 High</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Contains Upper 32 bits of EU Metrics, Event10</td>
</tr>
</tbody>
</table>
## EU Metrics for Event10 Low

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>138180h</td>
</tr>
</tbody>
</table>

Lower 14 bits of the EU Metrics Event10

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:18</td>
<td><strong>EUMetrics, Event10 Low</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Contains lower 14 bits of EU Metrics, Event10</td>
</tr>
<tr>
<td>17:0</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
</tbody>
</table>
## EU Metrics for Event11 High

<table>
<thead>
<tr>
<th>EUMETRICSEVENT11H - EU Metrics for Event11 High</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 13818Ch</td>
</tr>
</tbody>
</table>

Upper 32 bits of the EU Metrics Event11

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>EUMetrics, Event11 High</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Contains Upper 32 bits of EU Metrics, Event11</td>
</tr>
</tbody>
</table>
## EU Metrics for Event11 Low

### EUMETRICSEVENT11L - EU Metrics for Event11 Low

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>138188h</td>
</tr>
</tbody>
</table>

Lower 14 bits of the EU Metrics Event11

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:18</td>
<td><strong>EUMetrics, Event11 Low</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Contains lower 14 bits of EU Metrics, Event11</td>
</tr>
<tr>
<td>17:0</td>
<td>0000h</td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>17:0</td>
<td>0000h</td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
</tbody>
</table>
## EU Metrics for Event12 High

### EUMETRICSEVENT12H - EU Metrics for Event12 High

<table>
<thead>
<tr>
<th>Description</th>
<th>DWord</th>
<th>Bit</th>
<th>Default Value</th>
<th>Access</th>
<th>Contains Upper 32 bits of EU Metrics, Event12</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>EUMetrics, Event12 High</strong></td>
<td>0</td>
<td>31:0</td>
<td>00000000h</td>
<td>RO</td>
<td></td>
</tr>
</tbody>
</table>

- **Register Space**: MMIO: 0/2/0
- **Project**: CHV, BSW
- **Source**: PRM
- **Default Value**: 0x00000000
- **Size (in bits)**: 32
- **Address**: 138194h

Upper 32 bits of the EU Metrics Event12.
## EU Metrics for Event12 Low

<table>
<thead>
<tr>
<th>EUMETRICSEVENT12L - EU Metrics for Event12 Low</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 138190h</td>
</tr>
</tbody>
</table>

Lower 14 bits of the EU Metrics Event12

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:18</td>
<td><strong>EUMetrics, Event12 Low</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 0000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Contains lower 14 bits of EU Metrics, Event12</td>
</tr>
<tr>
<td>17:0</td>
<td>0</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> RO</td>
</tr>
</tbody>
</table>

Reserved
Event selection and base counters

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td>Counter 7 client</td>
</tr>
</tbody>
</table>

- **Access:** R/W
- **Incf_lpf_cnt7_client[7:0].Client Encoding (hex):**
  - GAFS Rd 00
  - GAFS Wr 01
  - HDC0 Data Rd 02
  - HDC0 Const Rd 03
  - HDC0 URB Rd 04
  - HDC0 Data Wr 05
  - HDC0 URB Wr 06
  - HDC1 Data Rd 07
  - HDC1 Const Rd 08
  - HDC1 URB Rd 09
  - HDC1 Data Wr 0A
  - HDC1 URB Wr 0B
  - TDL0 Rd 0C
  - TDL1 Rd 0D
  - Tex0 Rd 0E
  - Tex1 Rd 0F
  - Tex2 Rd (reserved) 10
  - Tex3 Rd (reserved) 11
  - SBE Rd 12
  - IC0 Rd 13
  - IC1 Rd 14
  - SARB Rd 15
  - Aggregated Tex 16
  - SLM0 Rd 17
  - SLM1 Rd 18
  - SLM0 Wr 19
  - SLM1 Wr 1A
  - SLM0 Atomics 1B
  - SLM1 Atomics 1C
  - Reserved 1D
  - Reserved 1E
<table>
<thead>
<tr>
<th><strong>LPFCREG2 - Event selection and base counters</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Reserved 1F</td>
</tr>
<tr>
<td>FF Stalls 20</td>
</tr>
<tr>
<td>HDC Stalls 21</td>
</tr>
<tr>
<td>TDL Stalls 22</td>
</tr>
<tr>
<td>Texture Stalls 23</td>
</tr>
<tr>
<td>IC Stalls 24</td>
</tr>
<tr>
<td>SBE Stalls 25</td>
</tr>
<tr>
<td>SLM Stalls 26</td>
</tr>
<tr>
<td>Bank0 Total Hits 40</td>
</tr>
<tr>
<td>Bank0 Total Cycles 41</td>
</tr>
<tr>
<td>Bank0 Total Rds 42</td>
</tr>
<tr>
<td>Bank0 Total Wrs 43</td>
</tr>
<tr>
<td>Bank0 FF Rds 44</td>
</tr>
<tr>
<td>Bank0 FF Wrs 45</td>
</tr>
<tr>
<td>Bank0 DC Rds 46</td>
</tr>
<tr>
<td>Bank0 DC Wrs 47</td>
</tr>
<tr>
<td>Bank0 DC Hits 48</td>
</tr>
<tr>
<td>rsvd 49</td>
</tr>
<tr>
<td>Bank0 Tex Rds 4A</td>
</tr>
<tr>
<td>Bank0 Tex Hits 4B</td>
</tr>
<tr>
<td>Bank0 IC Rds 4C</td>
</tr>
<tr>
<td>Bank0 IC Hits 4D</td>
</tr>
<tr>
<td>Reserved 4E</td>
</tr>
<tr>
<td>Reserved 4F</td>
</tr>
<tr>
<td>Bank1 Events 50-5F (except 59-reserved)</td>
</tr>
<tr>
<td>Bank2 Events 60-6F (except 69-reserved)</td>
</tr>
<tr>
<td>Bank3 Events 70-7F (except 79-reserved)</td>
</tr>
<tr>
<td>MSC Rd 80</td>
</tr>
<tr>
<td>MSC Wr 81</td>
</tr>
<tr>
<td>STC Rd 82</td>
</tr>
<tr>
<td>STC Wr 83</td>
</tr>
<tr>
<td>Hiz Rd 84</td>
</tr>
<tr>
<td>Hiz Wr 85</td>
</tr>
<tr>
<td>RCZ Rd 86</td>
</tr>
<tr>
<td>RCZ Wr 87</td>
</tr>
<tr>
<td>RCC Rd 88</td>
</tr>
<tr>
<td>RCC Wr 89</td>
</tr>
<tr>
<td>LTCD0 Err Corr EE</td>
</tr>
<tr>
<td>LTCD1 Err Corr EF</td>
</tr>
<tr>
<td>LTCD2 Err Corr F0</td>
</tr>
<tr>
<td>LTCD3 Err Corr F1</td>
</tr>
<tr>
<td>LTCD0 Err UnCorr F2</td>
</tr>
<tr>
<td>LTCD1 Err UnCorr F3</td>
</tr>
<tr>
<td>LTCD2 Err UnCorr F4</td>
</tr>
<tr>
<td>LTCD3 Err UnCorr F5</td>
</tr>
</tbody>
</table>
### LPFCREG2 - Event selection and base counters

<table>
<thead>
<tr>
<th>Counter#7 Client Selection: This field controls which client’s request stream is observed in counter#7.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>23:16</strong> Counter 6 client</td>
</tr>
<tr>
<td>Access: R/W</td>
</tr>
<tr>
<td>lncf_lpf_cnt6_client[7:0]</td>
</tr>
</tbody>
</table>

Counter#6 Client Selection: This field controls which client’s request stream is observed in counter#6.

<table>
<thead>
<tr>
<th>Counter#5 Client Selection: This field controls which client’s request stream is observed in counter#5.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>15:8</strong> Counter 5 client</td>
</tr>
<tr>
<td>Access: R/W</td>
</tr>
<tr>
<td>Incl_lpfc_cnt5_client[7:0]</td>
</tr>
</tbody>
</table>

Counter#5 Client Selection: This field controls which client’s request stream is observed in counter#5.

<table>
<thead>
<tr>
<th>Counter#4 Client Selection: This field controls which client’s request stream is observed in counter#4.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>7:0</strong> Counter 4 client</td>
</tr>
<tr>
<td>Access: R/W</td>
</tr>
<tr>
<td>Incl_lpfc_cnt4_client[7:0]</td>
</tr>
</tbody>
</table>
## Event Selection and Base Counters1

### LPFCREG1 - Event Selection and Base Counters1

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0B010h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>Counter 3 client</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Incf_lpfc_cnt3_client[7:0].Counter#3 Client Selection: This field controls which client’s request stream is observed in counter#3.</td>
</tr>
<tr>
<td>23:16</td>
<td></td>
<td><strong>Counter 2 client</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Incf_lpfc_cnt2_client[7:0].Counter#2 Client Selection: This field controls which client’s request stream is observed in counter#2.</td>
</tr>
<tr>
<td>15:8</td>
<td></td>
<td><strong>Counter 1 Client</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Incf_lpfc_cnt1_client[7:0].Counter#1 Client Selection: This field controls which client’s request stream is observed in counter#1.</td>
</tr>
<tr>
<td>7:0</td>
<td></td>
<td><strong>Counter0 Client</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Incf_lpfc_cnt0_client[7:0].Counter#0 Client Selection: This field controls which client’s request stream is observed in counter#0.</td>
</tr>
</tbody>
</table>
### EXECLIST_STATUS - Execlist Status

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000, 0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>RO</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>64</td>
</tr>
<tr>
<td>Address:</td>
<td>02234h-0223Bh</td>
</tr>
<tr>
<td>Name:</td>
<td>RCS Execlist Status</td>
</tr>
<tr>
<td>ShortName:</td>
<td>EXECLIST_STATUS_RCSUNIT</td>
</tr>
</tbody>
</table>

| Address:        | 12234h-1223Bh |
| Name:           | RCS Execlist Status |
| ShortName:      | EXECLIST_STATUS_VCSUNIT0 |

| Address:        | 1A234h-1A23Bh |
| Name:           | RCS Execlist Status |
| ShortName:      | EXECLIST_STATUS_VECSUNIT |

| Address:        | 1C234h-1C23Bh |
| Name:           | RCS Execlist Status |
| ShortName:      | EXECLIST_STATUS_VCSUNIT1 |

| Address:        | 22234h-2223Bh |
| Name:           | RCS Execlist Status |
| ShortName:      | EXECLIST_STATUS_BCSUNIT |

This register contains the pointers and full indicator for the Execlist Queue and the context ID of the currently running context. DefaultValue = UUUU UUU1h (4:0 default to 00001b, others UNDEFINED).

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td><strong>Current Context ID</strong></td>
</tr>
<tr>
<td></td>
<td>63:32</td>
<td>Format: U32</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Contains the context ID of the currently running context.</td>
</tr>
<tr>
<td>31:30</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td>29:27</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>31:30</td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td>29:27</td>
<td>Format: MBZ</td>
</tr>
</tbody>
</table>
### EXECLIST_STATUS - Execlist Status

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>26:19 Reserved</td>
<td>Project: CHV, BSW&lt;br&gt;Format: MBZ</td>
</tr>
<tr>
<td>18 Reserved</td>
<td>Project: CHV, BSW&lt;br&gt;Format: MBZ</td>
</tr>
<tr>
<td>17 Reserved</td>
<td>Project: CHV, BSW&lt;br&gt;Format: MBZ</td>
</tr>
<tr>
<td>16 Arbitration Enable</td>
<td>Project: CHV, BSW&lt;br&gt;Format: U1</td>
</tr>
<tr>
<td></td>
<td>This field reflects the Arbitration Flag set by the MI_ARB_ON_OFF command in Command Streamer.</td>
</tr>
<tr>
<td>15:14 Current Active Element Status</td>
<td>Project: CHV, BSW&lt;br&gt;Format: U2</td>
</tr>
<tr>
<td></td>
<td>Points at the element being executed in current Execlist (if there is one).</td>
</tr>
<tr>
<td>Value</td>
<td>Name</td>
</tr>
<tr>
<td>00b</td>
<td>No Active Element being executed</td>
</tr>
<tr>
<td>01b</td>
<td>Element0 of current execlist being executed</td>
</tr>
<tr>
<td>10b</td>
<td>Element1 of current execlist being executed</td>
</tr>
<tr>
<td>11b</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td>This field contains the switch reason for the last context to switch away, as captured in the Context Status Dword, bits 8:0.</td>
</tr>
<tr>
<td>Programming Notes</td>
<td>This field should not written by SW.</td>
</tr>
<tr>
<td>4 Execlist 0 Valid</td>
<td>Project: CHV, BSW&lt;br&gt;Format: Flag</td>
</tr>
<tr>
<td></td>
<td>This bit is set when the first DW for this Execlist port 0 is written through the submission port, and will not be cleared till the CSB is updated and the command stream is switching to the next execution list. If no execution list is pending, the transition of this bit from one to zero guarantees there will be no preemption on the next submission.</td>
</tr>
<tr>
<td>Value</td>
<td>Name</td>
</tr>
<tr>
<td>0</td>
<td>Invalid [Default]</td>
</tr>
<tr>
<td>1</td>
<td>Valid</td>
</tr>
<tr>
<td>3 Execlist 1 Valid</td>
<td>Project: CHV, BSW&lt;br&gt;Format: Flag</td>
</tr>
</tbody>
</table>

**Doc Ref # IHD-OS-CHV-BSW-Vol 2c-10.15**
EXECLIST_STATUS - Execlist Status

**Project:** CHV, BSW  
**Format:** Flag

This bit is set when the first DW for this Execlist port 1 is written through the submission port, and will not be cleared till the CSB is updated and the command stream is switching to the next execution list. If no execution list is pending, the transition of this bit from one to zero guarantees there will be no preemption on the next submission.

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Invalid</td>
</tr>
<tr>
<td>1</td>
<td>Valid</td>
</tr>
</tbody>
</table>

**Execlist Queue Full**

**Project:** CHV, BSW  
When [Execlist Write Pointer] and [Current Execlist Pointer] are equal, this bit differentiates between Queue Full and Queue Empty.

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Execlist Queue Empty</td>
<td>[Default]</td>
</tr>
<tr>
<td>1</td>
<td>Execlist Queue Full</td>
<td>There is a current and a pending execlist.</td>
</tr>
</tbody>
</table>

**Execlist Write Pointer**

**Project:** CHV, BSW  
**Format:** ExeclistContentsIndex

Determines which Execlist will be the next submitted to. When a new execlist is submitted, this pointer increments to point to the next execlist slot.

**Current Execlist Pointer**

**Default Value:** 1h  
**Project:** CHV, BSW  
**Format:** ExeclistContentsIndex

Points at the currently executing Execlist (if there is one). This pointer advances when the first context of new execlist is restored.
**Execlist Submit Port Register**

<table>
<thead>
<tr>
<th><strong>EXECLIST_SUBMITPORT</strong> - Execlist Submit Port Register</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Access:</strong> WO</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 02230h-02233h</td>
</tr>
<tr>
<td><strong>Name:</strong> Execlist Submit Port Register</td>
</tr>
<tr>
<td><strong>ShortName:</strong> EXECLIST_SUBMITPORT_RCSUNIT</td>
</tr>
</tbody>
</table>

| **Address:** 12230h-12233h                              |
| **Name:** Execlist Submit Port Register                 |
| **ShortName:** EXECLIST_SUBMITPORT_VCSUNIT0             |

| **Address:** 1A230h-1A233h                              |
| **Name:** Execlist Submit Port Register                 |
| **ShortName:** EXECLIST_SUBMITPORT_VCSUNIT              |

| **Address:** 1C230h-1C233h                              |
| **Name:** Execlist Submit Port Register                 |
| **ShortName:** EXECLIST_SUBMITPORT_VECSUNIT             |

| **Address:** 22230h-22233h                              |
| **Name:** Execlist Submit Port Register                 |
| **ShortName:** EXECLIST_SUBMITPORT_BCSUNIT              |

SW should submit a new pending execlist to this register. The DWs of the context descriptors must be written in a specific order: Element 1 must be written first and then Element 0. For each Element, DW1 must be written first followed by DW0. Context descriptors for both the elements must be written even if only one context are being submitted. The valid bits of the unused context descriptors should be set to 0.

**Order of DW Submission to the Execlist Port**

- Element 1, High Dword
- Element 1, Low Dword
- Element 0, High Dword
- Element 0, Low Dword

If a execlist of only one element is being submitted, it must be submitted in Element 0. It is UNDEFINED to submit a execlist with the valid bit of Element 0 clear (an “empty” execlist). It is possible that one or all of the contexts submitted in a execlists are “empty”; that is, have head and tail pointers equal to each other indicating no commands to be run. All of the valid bits in the Execlist Element Status Registers for the “about to be submitted” execlist will be cleared when the first DW (DW1 of Element 1) is written to the submit port.
EXECLIST_SUBMITPORT - Execlist Submit Port Register

Submission of the Element 0 Context Descriptor low Dword with the valid bit set is interpreted as a request to switch (as soon as possible) to the new execlist, i.e., a pre-emption request. If a submitted Execlist’s Element 0 Context Descriptor LRCA matches the LRCA of the currently executing context, then the newly submitted execlist will become the currently executing execlist without any context switch and without any impact to the executing context except that it will re-sample the tail pointer from the context image. This is done in case more commands have been inserted into its ring buffer between the first execlist submission and the 2nd.

<table>
<thead>
<tr>
<th>Programming Notes</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>SW must ensure the contexts submitted to the both the context descriptors in the execlist are different, i.e SW must not submit the same context descriptor to both the elements of the execlist.</td>
<td></td>
</tr>
<tr>
<td>SW must follow below programming sequence for ELSP submission in host mode of. SW must set Force Wakeup bit to prevent GT from entering C6 while ELSP writes are in progress. Ex: Set Force Wakeup Program ELSP writes Reset Force Wakeup</td>
<td>RenderCS</td>
</tr>
</tbody>
</table>

**Render CS Only:** Command Streamer triggers IDLE sequence flows for RDOP_CG on un-successful Semaphore Waits and Wait for Display Events when "Inhibit Synchronous Context Switch" is set in CTXT_SR_CTL register. As part of IDLE flows CS flushes the Write Caches (Z, Color, HDC). While IDLE flush in progress context switch can happen due to pending execlist submitted and when this condition occurs CS might not issue context switch flush resulting in RO caches not invalidated (State, Texture, Instruction, Constant).

To WA above issue, SW must always program a valid BB_PER_CTX_PTR for every context submitted with a PIPE_CONTROL command to invalidate all RO caches.

PIPE_CONTROL with below bits set in it:

- Instruction Cache Invalidate Enable
- Texture Cache Invalidate Enable
- Constant Cache Invalidate Enable
- State Cache Invalidate Enable

[All Command Streamers]: When SW intends to use semaphore signaling between Command streamers, SW must avoid lite restores in HW by programming Force Restore bit to ‘1’ in context descriptor during context submission, this is required to avoid known HW issue.

<table>
<thead>
<tr>
<th>Workaround</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Workaround: SW must always ensure there are valid commands to be executed by HW on a context submission, i.e ring buffer head pointer must not be equal to the ring buffer head pointer on context submission to HW for execution.</td>
<td>RenderCS</td>
</tr>
</tbody>
</table>
**EXECLIST_SUBMITPORT - Execlist Submit Port Register**

**Additional Note:**

This WA need not be applied when the arbitration is not disabled prior to executing “Batch Buffer Per Context Pointer” as part of context restore. Arbitration can be disabled prior to executing “Batch Buffer Per Context Pointer” by programming MI_ARB_ON_OFF (arbitration disable) in indirect context pointer.

**Workaround:**

SW must always ensure a preempted context submitted to HW doesn’t undergo lite restore due to the same context getting submitted on the next Execlist submission. This can be achieved by setting “Force Restore Bit” in the context descriptor of the context getting submitted and if the same context is known to be submitted to HW for execution on the earlier Execlist submission. 

**Or**

SW on submitting a preempted context must wait for the context to switch out before submitting the same context to the Execlist Submit Port.

**Note:**

This WA need not be applied when “Force Sync Command Ordering” bit of INSTPM register is not disabled (programmed to value ‘0’) during execution of “Batch Buffer Per Context Pointer” during context restore. “Force Sync Command Ordering” can be disabled prior to or during execution of “Batch Buffer Per Context Pointer” by programming INSTPM register using MI_LOAD_REGISTER_IMM command in Indirect Context Pointer or in “Batch Buffer Per Context Pointer”.

Disabling of “Force Sync Command Ordering” during “Batch Buffer Per Context Pointer” execution was required to address Resource Streamer related preemption issue on HSD 1912487, this WA is not applied when Resource Streamer (RS) is not enabled or when a Resource Streamer enabled context is not preemptable.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Context Descriptor DW</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: Context Descriptor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>See “Context Descriptor Format” for format. The element that this DW is submitted as and whether it is the high DW or the low DW is determined by order. This register must be written to 4 times in order to submit a execlist.</td>
</tr>
</tbody>
</table>
## Execute Condition Code Register

<table>
<thead>
<tr>
<th>EXCC - Execute Condition Code Register</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: RenderCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Access: R/W, RO</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Trusted Type: 1</td>
</tr>
<tr>
<td>Address: 02028h</td>
</tr>
<tr>
<td>Address: 12028h-1202Bh</td>
</tr>
<tr>
<td>Name: Execute Condition Code Register</td>
</tr>
<tr>
<td>ShortName: EXCC_VCSUNIT0</td>
</tr>
<tr>
<td>Address: 1A028h-1A02Bh</td>
</tr>
<tr>
<td>Name: Execute Condition Code Register</td>
</tr>
<tr>
<td>ShortName: EXCC_VECSUNIT</td>
</tr>
<tr>
<td>Address: 1C028h-1C02Bh</td>
</tr>
<tr>
<td>Name: Execute Condition Code Register</td>
</tr>
<tr>
<td>ShortName: EXCC_VCSUNIT1</td>
</tr>
<tr>
<td>Address: 22028h-2202Bh</td>
</tr>
<tr>
<td>Name: Execute Condition Code Register</td>
</tr>
<tr>
<td>ShortName: EXCC_BCSUNIT</td>
</tr>
</tbody>
</table>

This register contains user defined and hardware generated conditions that are used by MI_WAIT_FOR_EVENT commands. An MI_WAIT_FOR_EVENT instruction excludes the executing ring from arbitration if the selected event evaluates to a 1, while instruction is discarded if the condition evaluates to a 0. Once excluded a ring is enabled into arbitration when the selected condition evaluates to a 0.

This register also contains control for the invalidation of indirect state pointers on context restore.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td>Mask Bits</td>
</tr>
<tr>
<td></td>
<td>Format:</td>
<td>Mask[15:0]</td>
</tr>
<tr>
<td></td>
<td>These bits serves as a write enable for bits 15:0. If this register is written with any of these bits clear the corresponding bit in the field 15:0 will not be modified. Reading these bits always returns 0s.</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td></td>
<td>Format:</td>
<td>MBZ</td>
</tr>
</tbody>
</table>
## EXCC - Execute Condition Code Register

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Project</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td><strong>Context Wait for V-blank on Pipe-C</strong>&lt;br&gt;This field when set indicates the corresponding context has executed MI_WAIT_FOR_EVENT with &quot;Display Pipe C Vertical Blank Wait Enable&quot; set. This is an internal HW flag and should not be accessed by SW.</td>
<td>CHV, BSW</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td><strong>Context Wait for V-blank on Pipe-B</strong>&lt;br&gt;This field when set indicates the corresponding context has executed MI_WAIT_FOR_EVENT with &quot;Display Pipe B Vertical Blank Wait Enable&quot; set. This is an internal HW flag and should not be accessed by SW.</td>
<td>CHV, BSW</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td><strong>Context Wait for V-blank on Pipe-A</strong>&lt;br&gt;This field when set indicates the corresponding context has executed MI_WAIT_FOR_EVENT with &quot;Display Pipe A Vertical Blank Wait Enable&quot; set. This is an internal HW flag and should not be accessed by SW.</td>
<td>CHV, BSW</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td><strong>Pending Indirect State Dirty Bit</strong>&lt;br&gt;This field keeps track of whether or not an indirect state pointer command has been parsed in the current context. Clears either on a context save or explicitly through a flush command. This bit is Read Only.</td>
<td>CHV, BSW</td>
<td>RO</td>
</tr>
<tr>
<td>10:7</td>
<td><strong>Pending Indirect State Counter</strong>&lt;br&gt;This field keeps track of the maximum number of indirect state pointers pending in the system. When the register is saved/restored, it saves either a value of 1 or 0. This field is Read-Only.</td>
<td>CHV, BSW</td>
<td></td>
</tr>
<tr>
<td>6:5</td>
<td><strong>Reserved</strong>&lt;br&gt;Format: MBZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4:0</td>
<td><strong>User Defined Condition Codes</strong>&lt;br&gt;The software may signal a Stream Semaphore by setting the Mask bit and Signal Bit together to match the bit field specified in a WAIT_FOR_EVENT (Semaphore).</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## FAULT_TLB_RD_DATA0 Register

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>FAULT_TLB_READ_DATA0 Register</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fault cycle Virtual address [43:12]</td>
</tr>
</tbody>
</table>

Register Space: MMIO: 0/2/0
Project: CHV, BSW
Source: PRM
Default Value: 0x00000000
Size (in bits): 32
Address: 04B10h
## FAULT_TLB_RD_DATA1 Register

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>FAULT_TLB_READ_DATA1 Register</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[31:5] Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[3:0] Fault cycle Virtual address [47:44]</td>
</tr>
</tbody>
</table>
Fault Switch Out

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space</td>
<td>MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project</td>
<td>CHV</td>
</tr>
<tr>
<td>Source</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits)</td>
<td>32</td>
</tr>
<tr>
<td>Address</td>
<td>04590h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>Fault Switch Out</td>
<td>00000000h</td>
<td>R/W</td>
</tr>
</tbody>
</table>
## FBC_RT_BASE_ADDR_REGISTER

### FBC_RT_BASE_ADDR_REGISTER - FBC_RT_BASE_ADDR_REGISTER

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>RenderCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>07020h</td>
</tr>
<tr>
<td>Valid Projects:</td>
<td>CHV, BSW</td>
</tr>
</tbody>
</table>

This Register is saved and restored as part of Context.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:12</td>
<td><strong>FBC RT Base Address</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: PPGraphicsAddress[31:12]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4KB aligned Base Address as mapped in the PPGTT or in the GGTT for the render target. This base address must be the one that is either front buffer or the back-buffer (a flip target). It must be programmed before any draw call binding that render target base address.</td>
</tr>
<tr>
<td>11:2</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Format: PBC</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td><strong>FBC Front Buffer Target</strong></td>
</tr>
<tr>
<td></td>
<td>Project: CHV, BSW</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Format: Enable</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>[Default]</td>
<td>FBC is targeting the Back Buffer for compression. This buffer can be cached in the MLC/LLC, so a GFDT flush is required before FBC can begin compression.</td>
</tr>
<tr>
<td>1h</td>
<td></td>
<td>FBC is targeting the Font Buffer for compression. This buffer cannot be cached in the MLC/LLC. FBC compression can begin after any RC flush.</td>
</tr>
<tr>
<td>Value</td>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>0h</td>
<td>[Default]</td>
<td>Base address in this register [31:12] is not valid and therefore FBC will not get any modifications from rendering.</td>
</tr>
<tr>
<td>1h</td>
<td>PPGTT Render Target Base Address Valid for FBC</td>
<td>Base address in this register [31:12] is valid and HW needs to compare the current render target base address with this base address to provide modifications to FBC.</td>
</tr>
</tbody>
</table>
### FBC_RT_BASE_ADDR_REGISTER_UPPER - FBC_RT_BASE_ADDR_REGISTER_UPPER

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>RenderCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>07024h</td>
</tr>
</tbody>
</table>

This Register is saved and restored as part of Context.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format:</strong> PBC</td>
</tr>
<tr>
<td>15:0</td>
<td></td>
<td><strong>FBC RT Base Address High</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format:</strong> BaseAddress[47:32]</td>
</tr>
</tbody>
</table>

Must be set to modify corresponding data bit. Reads to this field returns zero. Upper 4KB aligned Base Address as mapped in the PPGTT or in the GGTT for the render target. This base address must be the one that is either front buffer or the back-buffer (a flip target). It can be only programmed once per context.

**Programming Notes**

It must be programmed before any draw call binding that render target base address.
FD

FD - FD

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>PCI: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>000C4h</td>
</tr>
</tbody>
</table>

Functional Disable. This register is used by SBIOS, not by driver.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:1</td>
<td>RESERVED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>FUNCTION_DISABLE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default Value: 0b</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
<td></td>
</tr>
</tbody>
</table>

Description

FD:
0 : Default - normal operation.
1 : When set, the function is disabled (configuration space is disabled). All new requests on the IOSF Primary bus, including any new configuration cycle requests are not claimed on IOSF Primary. This bit has no effect register accessibility via IOSF SB. Once programmed to '1', the only way to re-enable device 2 is via an IOSF SB write of '0' to this register.

CHV, BSW: Wire is sent to PSF for decode purposes gvd_psf_dev2disable_nczfwoh.
## FF Performance

### FF_PERF - FF Performance

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>RenderCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Trusted Type:</td>
<td>1</td>
</tr>
<tr>
<td>Address:</td>
<td>06B1Ch</td>
</tr>
<tr>
<td>Valid Projects:</td>
<td>CHV, BSW</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td><strong>Mask</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: All</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: WO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: Mask[15:0]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Must be set to modify corresponding bit in Bits 15:0. (All implemented bits)</td>
</tr>
<tr>
<td>15:11</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: All</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: r/w</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: PBC</td>
</tr>
<tr>
<td>10:8</td>
<td></td>
<td><strong>Throttle counter value</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: r/w</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: Disable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Counter value defining how many clocks the interface needs to be slowed down.</td>
</tr>
<tr>
<td>7:3</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: r/w</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: PBC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Value</td>
</tr>
<tr>
<td>0h</td>
<td>[Default]</td>
<td>Masked by default.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>[Default]</td>
<td>Masked by default.</td>
</tr>
</tbody>
</table>
### FF_PERF - FF Performance

<table>
<thead>
<tr>
<th></th>
<th>Enable throttling for SF-WM interface</th>
<th></th>
<th>Enable throttling for SF-SBE interface</th>
<th></th>
<th>Enable throttling for CL-SF interface</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Access:</strong></td>
<td>r/w</td>
<td><strong>Format:</strong></td>
<td>Disable</td>
<td><strong>Value</strong></td>
<td>Name</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td>0h</td>
<td>Disable</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1h</td>
<td>Enable</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>0h</td>
<td>Disable</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1h</td>
<td>Enable</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>0h</td>
<td>Disable</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1h</td>
<td>Enable</td>
</tr>
</tbody>
</table>
First Buffer Size and Start

**FBSS - First Buffer Size and Start**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0B420h</td>
</tr>
</tbody>
</table>

**LPFCREG02 - First Buffer Size and Start**

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td><strong>First Virtual Buffer Base</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>First Virtual Buffer Base: Programmed by driver to allocate a memory space for performance data storage. The buffer size should be aligned to the size of the memory allocated so it naturally aligns to the base (i.e. for 128KB bit[16]=0, 256KB bit[17:16]=0, 512KB bit[18:16]=0). Signal - lpconf_lpfc_virtual_base0 [31:16].</td>
</tr>
<tr>
<td>15:12</td>
<td>First Buffer Size</td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td>First Buffer Size: Determines the allowed buffer size for performance data storage. 0000b: 64KB. 0001b: 128KB. 0010b: 256KB. 0011b: 512KB. ... 1111b: 2GB. Signal - lpconf_lpfc_buffer_size0 [3:0].</td>
<td></td>
</tr>
<tr>
<td>11:3</td>
<td>Reserved</td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved.</td>
</tr>
</tbody>
</table>
### FBSS - First Buffer Size and Start

<table>
<thead>
<tr>
<th>2</th>
<th>Frame count and Draw call enable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
</tbody>
</table>

Enables the replacement of a specific L3 performance counter value in the reported data with a 16-bit tag created from the concatenation of the "Frame Count" and "Draw Call Number" programmable bitfields in the "Frame Count and Draw Call Number" register.

The exact counter replaced is dependent on the programmed value of the "Counter Enabling Selection" bitfield. The replaced counter is always the last one, except in the case only a single performance counter is enabled for reporting (in which no replacement occurs):

<table>
<thead>
<tr>
<th>CNTRENSEL Value</th>
<th>Replaced Event Counter</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>No Replacement</td>
</tr>
<tr>
<td>01</td>
<td>Counter 1</td>
</tr>
<tr>
<td>10</td>
<td>Counter 3</td>
</tr>
<tr>
<td>11</td>
<td>Counter 7</td>
</tr>
</tbody>
</table>

| 1 | Reserved |

<table>
<thead>
<tr>
<th>0</th>
<th>Master Counter Enable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
</tbody>
</table>

Master Counter Enable: This is the global enable for performance tracking. Once set, it kicks off all performance tracking mechanism.

Signal - lpconf_lpf_c_master_cnt_en.

This bit is used by all slices.
Flexible EU Event Control 0

<table>
<thead>
<tr>
<th>EU_PERF_CNT_CTL0 - Flexible EU Event Control 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Access: R/W</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 0E458h</td>
</tr>
</tbody>
</table>

This register configures flexible EU event 0/1. Please refer to the description of the flexible EU events for more details on supported events. Please note that this register is render context saved/restored.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: All</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
<tr>
<td>23:20</td>
<td></td>
<td><strong>Fine Event Filter Select EU event 1</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: U4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This field controls which fine event filter is applied to the coarsely filtered increment event when creating flexible EU event 1. Note that the fine event filter is logically applied after the coarse event filter.</td>
</tr>
<tr>
<td>19:16</td>
<td></td>
<td><strong>Coarse Event Filter Select EU event 1</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: U4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This field controls which coarse event filter is applied to the selected increment event when creating flexible EU event 1. Note that the coarse event filter is logically applied before the fine event filter.</td>
</tr>
<tr>
<td>15:12</td>
<td></td>
<td><strong>Increment Event for EU event 1</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: U4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This field controls which increment event provides the basis for flexible EU event 1.</td>
</tr>
<tr>
<td>11:8</td>
<td></td>
<td><strong>Fine Event Filter Select EU event 0</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: U4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This field controls which fine event filter is applied to the coarsely filtered increment event when creating flexible EU event 0. Note that the fine event filter is logically applied after the coarse event filter.</td>
</tr>
<tr>
<td>7:4</td>
<td></td>
<td><strong>Coarse Event Filter Select EU event 0</strong></td>
</tr>
</tbody>
</table>
## EU_PERF_CNT_CTL0 - Flexible EU Event Control 0

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Format</td>
<td>U4</td>
</tr>
</tbody>
</table>

This field controls which coarse event filter is applied to the selected increment event when creating flexible EU event 0. Note that the coarse event filter is logically applied before the fine event filter.

<table>
<thead>
<tr>
<th>3:0</th>
<th>Increment Event for EU event 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Format</td>
<td>U4</td>
</tr>
</tbody>
</table>

This field controls which increment event provides the basis for flexible EU event 0.
Flexible EU Event Control 1

**EU_PERF_CNT_CTL1 - Flexible EU Event Control 1**

- **Register Space:** MMIO: 0/2/0
- **Project:** CHV, BSW
- **Source:** PRM
- **Default Value:** 0x00000000
- **Access:** R/W
- **Size (in bits):** 32
- **Address:** 0E558h

This register configures flexible EU event 2/3. Please refer to the description of the flexible EU events for more details on supported events. Please note that this register is render context saved/-restored.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
<tr>
<td>23:20</td>
<td></td>
<td><strong>Fine Event Filter Select EU event 3</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: U4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This field controls which fine event filter is applied to the coarsely filtered increment event when creating flexible EU event 3. Note that the fine event filter is logically applied after the coarse event filter.</td>
</tr>
<tr>
<td>19:16</td>
<td></td>
<td><strong>Coarse Event Filter Select EU event 3</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: U4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This field controls which coarse event filter is applied to the selected increment event when creating flexible EU event 3. Note that the coarse event filter is logically applied before the fine event filter.</td>
</tr>
<tr>
<td>15:12</td>
<td></td>
<td><strong>Increment Event for EU event 3</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: U4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This field controls which increment event provides the basis for flexible EU event 3.</td>
</tr>
<tr>
<td>11:8</td>
<td></td>
<td><strong>Fine Event Filter Select EU event 2</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: U4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This field controls which fine event filter is applied to the coarsely filtered increment event when creating flexible EU event 2. Note that the fine event filter is logically applied after the coarse event filter.</td>
</tr>
</tbody>
</table>
### EU_PERF_CNT_CTL1 - Flexible EU Event Control 1

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Project</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>7:4</td>
<td>Coarse Event Filter Select EU event 2</td>
<td>CHV, BSW</td>
<td>U4</td>
</tr>
<tr>
<td></td>
<td>This field controls which coarse event filter is applied to the selected increment event when creating flexible EU event 2. Note that the coarse event filter is logically applied before the fine event filter.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3:0</td>
<td>Increment Event for EU event 2</td>
<td>CHV, BSW</td>
<td>U4</td>
</tr>
<tr>
<td></td>
<td>This field controls which increment event provides the basis for flexible EU event 2.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Flexible EU Event Control 2

<table>
<thead>
<tr>
<th>EU_PERF_CNT_CTL2 - Flexible EU Event Control 2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 0E658h</td>
</tr>
</tbody>
</table>

This register configures flexible EU event 4/5. Please refer to the description of the flexible EU events for more details on supported events. Please note that this register is render context saved/restored.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format:</strong> MBZ</td>
</tr>
<tr>
<td>23:20</td>
<td></td>
<td><strong>Fine Event Filter Select EU event 5</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format:</strong> U4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This field controls which fine event filter is applied to the coarsely filtered increment event when creating flexible EU event 5. Note that the fine event filter is logically applied after the coarse event filter.</td>
</tr>
<tr>
<td>19:16</td>
<td></td>
<td><strong>Coarse Event Filter Select EU event 5</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format:</strong> U4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This field controls which coarse event filter is applied to the selected increment event when creating flexible EU event 5. Note that the coarse event filter is logically applied before the fine event filter.</td>
</tr>
<tr>
<td>15:12</td>
<td></td>
<td><strong>Increment Event for EU event 5</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format:</strong> U4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This field controls which increment event provides the basis for flexible EU event 5.</td>
</tr>
<tr>
<td>11:8</td>
<td></td>
<td><strong>Fine Event Filter Select EU event 4</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format:</strong> U4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This field controls which fine event filter is applied to the coarsely filtered increment event when creating flexible EU event 4. Note that the fine event filter is logically applied after the coarse event filter.</td>
</tr>
</tbody>
</table>
**EU_PERF_CNT_CTL2 - Flexible EU Event Control 2**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>7:4</strong></td>
<td><strong>Coarse Event Filter Select EU event 4</strong></td>
</tr>
<tr>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td>Format: U4</td>
</tr>
<tr>
<td></td>
<td>This field controls which coarse event filter is applied to the selected increment event when creating flexible EU event 4. Note that the coarse event filter is logically applied before the fine event filter.</td>
</tr>
<tr>
<td><strong>3:0</strong></td>
<td><strong>Increment Event for EU event 4</strong></td>
</tr>
<tr>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td>Format: U4</td>
</tr>
<tr>
<td></td>
<td>This field controls which increment event provides the basis for flexible EU event 4.</td>
</tr>
</tbody>
</table>
Flexible EU Event Control 3

**EU_PERF_CNT_CTL3 - Flexible EU Event Control 3**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0E758h</td>
</tr>
</tbody>
</table>

This register configures flexible EU event 6/7. Please refer to the description of the flexible EU events for more details on supported events. Please note that this register is render context saved/ restored.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Project:</th>
<th>Format:</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>Reserved</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Project:</strong></td>
<td>All</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format:</strong></td>
<td>MBZ</td>
<td></td>
</tr>
<tr>
<td>23:20</td>
<td></td>
<td><strong>Fine Event Filter Select EU event 7</strong></td>
<td>CHV, BSW</td>
<td>U4</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Project:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>This field controls which fine event filter is applied to the coarsely filtered increment event when creating flexible EU event 7. Note that the fine event filter is logically applied after the coarse event filter.</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19:16</td>
<td></td>
<td><strong>Coarse Event Filter Select EU event 7</strong></td>
<td>CHV, BSW</td>
<td>U4</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Project:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>This field controls which coarse event filter is applied to the selected increment event when creating flexible EU event 7. Note that the coarse event filter is logically applied before the fine event filter.</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15:12</td>
<td></td>
<td><strong>Increment Event for EU event 7</strong></td>
<td>CHV, BSW</td>
<td>U4</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Project:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>This field controls which increment event provides the basis for flexible EU event 7.</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11:8</td>
<td></td>
<td><strong>Fine Event Filter Select EU event 6</strong></td>
<td>CHV, BSW</td>
<td>U4</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Project:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>This field controls which fine event filter is applied to the coarsely filtered increment event when creating flexible EU event 6. Note that the fine event filter is logically applied after the coarse event filter.</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### EU_PERF_CNT_CTL3 - Flexible EU Event Control 3

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Project</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>7:4</td>
<td>Coarse Event Filter Select EU event 6</td>
<td>CHV, BSW</td>
<td>U4</td>
</tr>
<tr>
<td></td>
<td>This field controls which coarse event filter is applied to the selected increment event when creating flexible EU event 6. Note that the coarse event filter is logically applied before the fine event filter.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3:0</td>
<td>Increment Event for EU event 6</td>
<td>CHV, BSW</td>
<td>U4</td>
</tr>
<tr>
<td></td>
<td>This field controls which increment event provides the basis for flexible EU event 6.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Flexible EU Event Control 4

### EU_PERF_CNT_CTL4 - Flexible EU Event Control 4

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0E45Ch</td>
</tr>
</tbody>
</table>

This register configures flexible EU event 8/9. Please refer to the description of the flexible EU events for more details on supported events. Please note that this register is render context saved/restore.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
<tr>
<td>23:20</td>
<td><strong>Fine Event Filter Select EU event 9</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Project: CHV, BSW</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Format: U4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>This field controls which fine event filter is applied to the coarsely filtered increment event when creating flexible EU event 9. Note that the fine event filter is logically applied after the coarse event filter.</td>
<td></td>
</tr>
<tr>
<td>19:16</td>
<td><strong>Coarse Event Filter Select EU event 9</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Project: CHV, BSW</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Format: U4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>This field controls which coarse event filter is applied to the selected increment event when creating flexible EU event 9. Note that the coarse event filter is logically applied before the fine event filter.</td>
<td></td>
</tr>
<tr>
<td>15:12</td>
<td><strong>Increment Event for EU event 9</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Project: CHV, BSW</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Format: U4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>This field controls which increment event provides the basis for flexible EU event 9.</td>
<td></td>
</tr>
<tr>
<td>11:8</td>
<td><strong>Fine Event Filter Select EU event 8</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Project: CHV, BSW</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Format: U4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>This field controls which fine event filter is applied to the coarsely filtered increment event when creating flexible EU event 8. Note that the fine event filter is logically applied after the coarse event filter.</td>
<td></td>
</tr>
<tr>
<td>7:4</td>
<td><strong>Coarse Event Filter Select EU event 8</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Project: CHV, BSW</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Format: U4</td>
<td></td>
</tr>
</tbody>
</table>
## EU_PERF_CNT_CTL4 - Flexible EU Event Control 4

<table>
<thead>
<tr>
<th></th>
<th>Project:</th>
<th>CHV, BSW</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Format:</td>
<td>U4</td>
</tr>
</tbody>
</table>

This field controls which coarse event filter is applied to the selected increment event when creating flexible EU event 8. Note that the coarse event filter is logically applied before the fine event filter.

<table>
<thead>
<tr>
<th></th>
<th>Increment Event for EU event 8</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Project:</td>
</tr>
<tr>
<td></td>
<td>Format:</td>
</tr>
</tbody>
</table>

This field controls which increment event provides the basis for flexible EU event 8.
**Flexible EU Event Control 5**

| **EU_PERF_CNT_CTL5 - Flexible EU Event Control 5** |
|---------------------------------|----------------|
| **Register Space:** MMIO: 0/2/0 | **Project:** CHV, BSW |
| **Source:** PRM | **Default Value:** 0x00000000 |
| **Access:** R/W | **Size (in bits):** 32 |
| **Address:** 0E55Ch | |

This register configures flexible EU event 10/11. Please refer to the description of the flexible EU events for more details on supported events. Please note that this register is render context saved/restored.

<table>
<thead>
<tr>
<th><strong>DWord</strong></th>
<th><strong>Bit</strong></th>
<th><strong>Reserved</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Dword</td>
<td>31:24</td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td><strong>Project:</strong></td>
<td>All</td>
<td><strong>Format:</strong> MBZ</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>23:20</strong></th>
<th><strong>Fine Event Filter Select EU event 11</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Project:</strong></td>
<td>CHV, BSW</td>
</tr>
<tr>
<td><strong>This field controls which fine event filter is applied to the coarsely filtered increment event when creating flexible EU event 11. Note that the fine event filter is logically applied after the coarse event filter.</strong></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>19:16</strong></th>
<th><strong>Coarse Event Filter Select EU event 11</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Project:</strong></td>
<td>CHV, BSW</td>
</tr>
<tr>
<td><strong>This field controls which coarse event filter is applied to the selected increment event when creating flexible EU event 11. Note that the coarse event filter is logically applied before the fine event filter.</strong></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>15:12</strong></th>
<th><strong>Increment Event for EU event 11</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Project:</strong></td>
<td>CHV, BSW</td>
</tr>
<tr>
<td><strong>This field controls which increment event provides the basis for flexible EU event 11.</strong></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>11:8</strong></th>
<th><strong>Fine Event Filter Select EU event 10</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Project:</strong></td>
<td>CHV, BSW</td>
</tr>
<tr>
<td><strong>This field controls which fine event filter is applied to the coarsely filtered increment event when creating flexible EU event 10. Note that the fine event filter is logically applied after the coarse event filter.</strong></td>
<td></td>
</tr>
</tbody>
</table>
### EU_PERF_CNT_CTL5 - Flexible EU Event Control 5

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Project</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>7:4</td>
<td><strong>Coarse Event Filter Select EU event 10</strong></td>
<td>CHV, BSW</td>
<td>U4</td>
</tr>
<tr>
<td></td>
<td>This field controls which coarse event filter is applied to the selected increment event when creating flexible EU event 10. Note that the coarse event filter is logically applied before the fine event filter.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3:0</td>
<td><strong>Increment Event for EU event 10</strong></td>
<td>CHV, BSW</td>
<td>U4</td>
</tr>
<tr>
<td></td>
<td>This field controls which increment event provides the basis for flexible EU event 10.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Flexible EU Event Control 6

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EU_PERF_CNT_CTL6 - Flexible EU Event Control 6</td>
<td></td>
</tr>
<tr>
<td>Register Space:</td>
<td>MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0E65Ch</td>
</tr>
</tbody>
</table>

This register configures flexible EU event 12/13. Please refer to the description of the flexible EU events for more details on supported events. Please note that this register is render context saved/restored.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td>Project:</td>
<td>All</td>
</tr>
<tr>
<td></td>
<td>Format:</td>
<td>MBZ</td>
</tr>
<tr>
<td>23:20</td>
<td><strong>Fine Event Filter Select EU event 13</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td></td>
<td>Format:</td>
<td>U4</td>
</tr>
<tr>
<td></td>
<td>This field controls which fine event filter is applied to the coarsely filtered increment event when creating flexible EU event 13. Note that the fine event filter is logically applied after the coarse event filter.</td>
<td></td>
</tr>
<tr>
<td>19:16</td>
<td><strong>Coarse Event Filter Select EU event 13</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td></td>
<td>Format:</td>
<td>U4</td>
</tr>
<tr>
<td></td>
<td>This field controls which coarse event filter is applied to the selected increment event when creating flexible EU event 13. Note that the coarse event filter is logically applied before the fine event filter.</td>
<td></td>
</tr>
<tr>
<td>15:12</td>
<td><strong>Increment Event for EU event 13</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td></td>
<td>Format:</td>
<td>U4</td>
</tr>
<tr>
<td></td>
<td>This field controls which increment event provides the basis for flexible EU event 13.</td>
<td></td>
</tr>
<tr>
<td>11:8</td>
<td><strong>Fine Event Filter Select EU event 12</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td></td>
<td>Format:</td>
<td>U4</td>
</tr>
<tr>
<td></td>
<td>This field controls which fine event filter is applied to the coarsely filtered increment event when creating flexible EU event 12. Note that the fine event filter is logically applied after the coarse event filter.</td>
<td></td>
</tr>
</tbody>
</table>
## EU_PERF_CNT_CTL6 - Flexible EU Event Control 6

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Project</th>
<th>Format</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>7:4 Coarse Event Filter Select EU event 12</td>
<td>This field controls which coarse event filter is applied to the selected increment event when creating flexible EU event 12. Note that the coarse event filter is logically applied before the fine event filter.</td>
<td>CHV, BSW</td>
<td>U4</td>
<td></td>
</tr>
<tr>
<td>3:0 Increment Event for EU event 12</td>
<td>This field controls which increment event provides the basis for flexible EU event 12.</td>
<td>CHV, BSW</td>
<td>U4</td>
<td></td>
</tr>
</tbody>
</table>
## FORCE_TO_NONPRIV

<table>
<thead>
<tr>
<th>FORCE_TO_NONPRIV - FORCE_TO_NONPRIV</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00002094</td>
</tr>
<tr>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 024D0h-024D3h</td>
</tr>
<tr>
<td><strong>Name:</strong> FORCE_TO_NONPRIV</td>
</tr>
<tr>
<td><strong>ShortName:</strong> FORCE_TO_NONPRIV_0_RCSUNIT</td>
</tr>
<tr>
<td><strong>Address:</strong> 024D4h-024D7h</td>
</tr>
<tr>
<td><strong>Name:</strong> FORCE_TO_NONPRIV</td>
</tr>
<tr>
<td><strong>ShortName:</strong> FORCE_TO_NONPRIV_1_RCSUNIT</td>
</tr>
<tr>
<td><strong>Address:</strong> 024D8h-024DBh</td>
</tr>
<tr>
<td><strong>Name:</strong> FORCE_TO_NONPRIV</td>
</tr>
<tr>
<td><strong>ShortName:</strong> FORCE_TO_NONPRIV_2_RCSUNIT</td>
</tr>
<tr>
<td><strong>Address:</strong> 024DCh-024DFh</td>
</tr>
<tr>
<td><strong>Name:</strong> FORCE_TO_NONPRIV</td>
</tr>
<tr>
<td><strong>ShortName:</strong> FORCE_TO_NONPRIV_3_RCSUNIT</td>
</tr>
<tr>
<td><strong>Address:</strong> 024E0h-024E3h</td>
</tr>
<tr>
<td><strong>Name:</strong> FORCE_TO_NONPRIV</td>
</tr>
<tr>
<td><strong>ShortName:</strong> FORCE_TO_NONPRIV_4_RCSUNIT</td>
</tr>
<tr>
<td><strong>Address:</strong> 024E4h-024E7h</td>
</tr>
<tr>
<td><strong>Name:</strong> FORCE_TO_NONPRIV</td>
</tr>
<tr>
<td><strong>ShortName:</strong> FORCE_TO_NONPRIV_5_RCSUNIT</td>
</tr>
<tr>
<td><strong>Address:</strong> 024E8h-024EBh</td>
</tr>
<tr>
<td><strong>Name:</strong> FORCE_TO_NONPRIV</td>
</tr>
<tr>
<td><strong>ShortName:</strong> FORCE_TO_NONPRIV_6_RCSUNIT</td>
</tr>
<tr>
<td><strong>Address:</strong> 024ECh-024EFh</td>
</tr>
<tr>
<td><strong>Name:</strong> FORCE_TO_NONPRIV</td>
</tr>
<tr>
<td><strong>ShortName:</strong> FORCE_TO_NONPRIV_7_RCSUNIT</td>
</tr>
<tr>
<td><strong>Address:</strong> 024F0h-024F3h</td>
</tr>
<tr>
<td><strong>Name:</strong> FORCE_TO_NONPRIV</td>
</tr>
<tr>
<td><strong>ShortName:</strong> FORCE_TO_NONPRIV_8_RCSUNIT</td>
</tr>
<tr>
<td>Address</td>
</tr>
<tr>
<td>--------------</td>
</tr>
<tr>
<td>024F4h-024F7h</td>
</tr>
<tr>
<td>024F8h-024FBh</td>
</tr>
<tr>
<td>024FCh-024FFh</td>
</tr>
<tr>
<td>124D0h-124D3h</td>
</tr>
<tr>
<td>124D4h-124D7h</td>
</tr>
<tr>
<td>124D8h-124DBh</td>
</tr>
<tr>
<td>124DCh-124DFh</td>
</tr>
<tr>
<td>124E0h-124E3h</td>
</tr>
<tr>
<td>124E4h-124E7h</td>
</tr>
<tr>
<td>124E8h-124EBh</td>
</tr>
<tr>
<td>124ECh-124EFh</td>
</tr>
<tr>
<td>124F0h-124F3h</td>
</tr>
<tr>
<td>ShortName: FORCE_TO_NONPRIV_8_VCSUNIT0</td>
</tr>
<tr>
<td>---------------------------------------</td>
</tr>
<tr>
<td>Address: 124F4h-124F7h</td>
</tr>
<tr>
<td>Name: FORCE_TO_NONPRIV</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ShortName: FORCE_TO_NONPRIV_9_VCSUNIT0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address: 124F8h-124FBh</td>
</tr>
<tr>
<td>Name: FORCE_TO_NONPRIV</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ShortName: FORCE_TO_NONPRIV_10_VCSUNIT0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address: 124FC8h-124FF7h</td>
</tr>
<tr>
<td>Name: FORCE_TO_NONPRIV</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ShortName: FORCE_TO_NONPRIV_0_VECSUNIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address: 1A4D4h-1A4D7h</td>
</tr>
<tr>
<td>Name: FORCE_TO_NONPRIV</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ShortName: FORCE_TO_NONPRIV_1_VECSUNIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address: 1A4D8h-1A4D7h</td>
</tr>
<tr>
<td>Name: FORCE_TO_NONPRIV</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ShortName: FORCE_TO_NONPRIV_2_VECSUNIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address: 1A4DCh-1A4DF7h</td>
</tr>
<tr>
<td>Name: FORCE_TO_NONPRIV</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ShortName: FORCE_TO_NONPRIV_3_VECSUNIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address: 1A4E0h-1A4E6h</td>
</tr>
<tr>
<td>Name: FORCE_TO_NONPRIV</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ShortName: FORCE_TO_NONPRIV_4_VECSUNIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address: 1A4E4h-1A4E7h</td>
</tr>
<tr>
<td>Name: FORCE_TO_NONPRIV</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ShortName: FORCE_TO_NONPRIV_5_VECSUNIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address: 1A4E8h-1A4EBh</td>
</tr>
<tr>
<td>Name: FORCE_TO_NONPRIV</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ShortName: FORCE_TO_NONPRIV_6_VECSUNIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address: 1A4EC8h-1A4EF7h</td>
</tr>
<tr>
<td>Name: FORCE_TO_NONPRIV</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ShortName: FORCE_TO_NONPRIV_7_VECSUNIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address: 1A4F0h-1A4F3h</td>
</tr>
<tr>
<td>Name: FORCE_TO_NONPRIV</td>
</tr>
<tr>
<td>Name:</td>
</tr>
<tr>
<td>--------------------</td>
</tr>
<tr>
<td>ShortName:</td>
</tr>
<tr>
<td>Address:</td>
</tr>
<tr>
<td>Name:</td>
</tr>
<tr>
<td>ShortName:</td>
</tr>
<tr>
<td>Address:</td>
</tr>
<tr>
<td>Name:</td>
</tr>
<tr>
<td>ShortName:</td>
</tr>
<tr>
<td>Address:</td>
</tr>
<tr>
<td>Name:</td>
</tr>
<tr>
<td>ShortName:</td>
</tr>
<tr>
<td>Address:</td>
</tr>
<tr>
<td>Name:</td>
</tr>
<tr>
<td>ShortName:</td>
</tr>
<tr>
<td>Address:</td>
</tr>
<tr>
<td>Name:</td>
</tr>
<tr>
<td>ShortName:</td>
</tr>
<tr>
<td>Address:</td>
</tr>
<tr>
<td>Name:</td>
</tr>
<tr>
<td>ShortName:</td>
</tr>
<tr>
<td>Address:</td>
</tr>
<tr>
<td>Name:</td>
</tr>
<tr>
<td>ShortName:</td>
</tr>
<tr>
<td>Address:</td>
</tr>
<tr>
<td>Name:</td>
</tr>
<tr>
<td>ShortName:</td>
</tr>
<tr>
<td>Address:</td>
</tr>
<tr>
<td>Name:</td>
</tr>
<tr>
<td>ShortName:</td>
</tr>
<tr>
<td>Address:</td>
</tr>
<tr>
<td>Name:</td>
</tr>
<tr>
<td>ShortName:</td>
</tr>
<tr>
<td>Address</td>
</tr>
<tr>
<td>-----------------</td>
</tr>
<tr>
<td>Name</td>
</tr>
<tr>
<td>ShortName</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Address</th>
<th>1C4F4h-1C4F7h</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>FORCE_TO_NONPRIV</td>
</tr>
<tr>
<td>ShortName</td>
<td>FORCE_TO_NONPRIV_9_VCSUNIT1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Address</th>
<th>1C4F8h-1C4FBh</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>FORCE_TO_NONPRIV</td>
</tr>
<tr>
<td>ShortName</td>
<td>FORCE_TO_NONPRIV_10_VCSUNIT1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Address</th>
<th>1C4FCh-1C4FFh</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>FORCE_TO_NONPRIV</td>
</tr>
<tr>
<td>ShortName</td>
<td>FORCE_TO_NONPRIV_11_VCSUNIT1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Address</th>
<th>224D0h-224D3h</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>FORCE_TO_NONPRIV</td>
</tr>
<tr>
<td>ShortName</td>
<td>FORCE_TO_NONPRIV_0_BCSUNIT</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Address</th>
<th>224D4h-224D7h</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>FORCE_TO_NONPRIV</td>
</tr>
<tr>
<td>ShortName</td>
<td>FORCE_TO_NONPRIV_1_BCSUNIT</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Address</th>
<th>224D8h-224DBh</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>FORCE_TO_NONPRIV</td>
</tr>
<tr>
<td>ShortName</td>
<td>FORCE_TO_NONPRIV_2_BCSUNIT</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Address</th>
<th>224DCh-224DFh</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>FORCE_TO_NONPRIV</td>
</tr>
<tr>
<td>ShortName</td>
<td>FORCE_TO_NONPRIV_3_BCSUNIT</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Address</th>
<th>224E0h-224E3h</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>FORCE_TO_NONPRIV</td>
</tr>
<tr>
<td>ShortName</td>
<td>FORCE_TO_NONPRIV_4_BCSUNIT</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Address</th>
<th>224E4h-224E7h</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>FORCE_TO_NONPRIV</td>
</tr>
<tr>
<td>ShortName</td>
<td>FORCE_TO_NONPRIV_5_BCSUNIT</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Address</th>
<th>224E8h-224EBh</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>FORCE_TO_NONPRIV</td>
</tr>
<tr>
<td>ShortName</td>
<td>FORCE_TO_NONPRIV_6_BCSUNIT</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Address</th>
<th>224ECH-224EFh</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>FORCE_TO_NONPRIV</td>
</tr>
</tbody>
</table>
# FORCE_TO_NONPRIV - FORCE_TO_NONPRIV

| ShortName: FORCE_TO_NONPRIV_7_BCSUNIT | Address: 224F0h-224F3h | Name: FORCE_TO_NONPRIV |
| ShortName: FORCE_TO_NONPRIV_8_BCSUNIT | Address: 224F4h-224F7h | Name: FORCE_TO_NONPRIV |
| ShortName: FORCE_TO_NONPRIV_9_BCSUNIT | Address: 224F8h-224FBh | Name: FORCE_TO_NONPRIV |
| ShortName: FORCE_TO_NONPRIV_10_BCSUNIT | Address: 224FC0h-224FFh | Name: FORCE_TO_NONPRIV |
| ShortName: FORCE_TO_NONPRIV_11_BCSUNIT | Address: 224F0h-224F3h | Name: FORCE_TO_NONPRIV |

These registers are privilege registers and are not allowed to be written from non-privilege batch buffer. These are global registers and power context save/restored.

## Programming Notes

RCS_FORCE_TO_NONPRIV registers in render CS must be used to force the below registers to be treated as non-privileged by HW:
- 0x7700 (GLOBAL_CLEAR_VALUE_0)
- 0x7704 (GLOBAL_CLEAR_VALUE_1)
- 0x7708 (GLOBAL_CLEAR_VALUE_2)
- 0x770C (GLOBAL_CLEAR_VALUE_3)

## DWord Bit Description

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:26</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td>Format:</td>
<td>MBZ</td>
</tr>
<tr>
<td>25:2</td>
<td></td>
<td>Non Privilege Register Address</td>
</tr>
<tr>
<td></td>
<td>Format:</td>
<td>MmioAddress[25:2]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This field contains the MMIO offset of a register. MMIO offset programmed in this field will be treated as a non-privilege register by render command streamer while processing register writes from a non-privilege batch buffer. This register provides programmability to extend the non-privilege register table mentioned in MI_BATCH_BUFFER_START command in render command streamer.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Value</td>
<td>Name</td>
</tr>
<tr>
<td>825h</td>
<td>1:0</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td>Format:</td>
<td>MBZ</td>
</tr>
</tbody>
</table>
Frame count and Draw call number

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>15:8</td>
<td>Frame Number</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Frame number is the first of two reporting tags that software (i.e. driver) may populate in order to provide reference points during L3 performance reporting modes. Should the &quot;Frame Count and Draw Call Enable&quot; bit (FCDCE) in the &quot;First Buffer Size and Start&quot; register be set, LPFC will selectively replace one of the reporting events with this programmable tag (in addition to the &quot;Draw Call Number&quot; field below). Software may use this to provide reference points for L3 performance counts when parsing the resulting data stream to align reported counts to higher-level operations. The original incarnation called for software to increment this value with each frame, however, the field is generic and may be used for any tagging purpose.</td>
<td></td>
</tr>
<tr>
<td>7:0</td>
<td>Draw call number</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td>The draw call number is the second programmable reporting tag provided by LPFC. With this second programmable tag, a more granular sampling boundary may be created by software, or it may be used to provide an alternative reference point for tracking L3 performance. The original incarnation called for software to increment this value with every draw call, but the field is generic and may be used for any similar purpose.</td>
<td></td>
</tr>
</tbody>
</table>
## FuseWord0

### FuseWord0 - FuseWord0

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>182130h</td>
</tr>
</tbody>
</table>

Fuse readout information.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Fusedw0</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fuse Information DW0</td>
</tr>
</tbody>
</table>
## FUSEWORD1

### FUSEWORD1 - FUSEWORD1

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>182134h</td>
</tr>
</tbody>
</table>

Fuse readout information.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>FUSEDW1</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fuse Information DW1</td>
</tr>
</tbody>
</table>
FUSEWORD2

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>FUSEDW2</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fuse Information DW2</td>
</tr>
<tr>
<td>DWord</td>
<td>Bit</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>-----</td>
<td>----------------------</td>
</tr>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>FUSEDW3</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fuse Information DW3</td>
</tr>
</tbody>
</table>
## FUSEWORD4

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space</td>
<td>MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits)</td>
<td>32</td>
</tr>
<tr>
<td>Address</td>
<td>182140h</td>
</tr>
</tbody>
</table>

Fuse readout information.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>FUSEDW4</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fuse Information DW4</td>
</tr>
</tbody>
</table>
## FUSEWORD5

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>182144h</td>
</tr>
</tbody>
</table>

Fuse readout information.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>FUSEDW5</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fuse Information DW5</td>
</tr>
</tbody>
</table>
## FUSEWORD6

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>182148h</td>
</tr>
</tbody>
</table>

Fuse readout information.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>FUSEDW6</td>
</tr>
</tbody>
</table>

**Default Value:** 00000000h

**Access:** RO

Fuse Information DW6
# FUSEWORD7

## FUSEWORD7 - FUSEWORD7

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>18214Ch</td>
</tr>
</tbody>
</table>

Fuse readout information.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>FUSEDW7</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fuse Information DW7</td>
</tr>
</tbody>
</table>
## FUSEWORD8

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space</td>
<td>MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits)</td>
<td>32</td>
</tr>
<tr>
<td>Address</td>
<td>182150h</td>
</tr>
</tbody>
</table>

Fuse readout information.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>FUSEDW8</strong></td>
</tr>
</tbody>
</table>

- **Default Value:** 00000000h
- **Access:** RO
- **Fuse Information DW8**
## FUSEWORD9

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>182154h</td>
</tr>
</tbody>
</table>

Fuse readout information.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>FUSEDW9</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fuse Information DW9</td>
</tr>
</tbody>
</table>

---

*Fuse Information DW9*
### FUSEWORD10 - FUSEWORD10

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>182158h</td>
</tr>
</tbody>
</table>

Fuse readout information.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>FUSEDW10</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: <strong>00000000h</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: <strong>RO</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fuse information DW10</td>
</tr>
</tbody>
</table>
### FUSEWORD11

#### Register Space:
MMIO: 0/2/0

#### Project:
CHV, BSW

#### Source:
PRM

#### Default Value:
0x00000000

#### Size (in bits):
32

#### Address:
18215Ch

Fuse readout information.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>FUSEDW11</strong></td>
</tr>
</tbody>
</table>

**Default Value:**
00000000h

**Access:**
RO

Fuse information DW11
## FUSEWORD12

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong></td>
<td>MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong></td>
<td>CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong></td>
<td>PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong></td>
<td>0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong></td>
<td>32</td>
</tr>
<tr>
<td><strong>Address:</strong></td>
<td>182160h</td>
</tr>
</tbody>
</table>

Fuse readout information. CHV, BSW: New for CHV, BSW.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>FUSEDW12</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fuse information DW12</td>
</tr>
</tbody>
</table>
# FUSEWORD13

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>182164h</td>
</tr>
</tbody>
</table>

Fuse readout information. CHV, BSW: New for CHV, BSW.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>FUSEDW13</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fuse information DW13</td>
</tr>
</tbody>
</table>
## Fuseword14

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>Fuseword14</td>
</tr>
</tbody>
</table>

**Default Value:**

00000000h

**Access:**

RO

Fuse information DW14

- [31:28] - FUSE_GT_EU_DISABLE: SS1, ROW1 - EU[3:0]
- [23:20] - FUSE_GT_EU_DISABLE: SS0, ROW1 - EU[3:0]
- [10] - FUSE_GT_SUBSLICE_DISABLE - SS0
**FUSEWORD15**

<table>
<thead>
<tr>
<th><strong>FUSEWORD15 - FUSEWORD15</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 18216Ch</td>
</tr>
</tbody>
</table>

Fuse readout information. CHV, BSW: New for CHV, BSW.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>FUSEDW15</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fuse information DW15</td>
</tr>
</tbody>
</table>
## FUSEWORD16

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x000000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>182170h</td>
</tr>
</tbody>
</table>

Fuse readout information. CHV, BSW: New for CHV, BSW.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>FUSEDW16</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: $00000000h$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fuse information DW16</td>
</tr>
</tbody>
</table>
## FUSEWORD17

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>182174h</td>
</tr>
</tbody>
</table>

Fuse readout information. CHV, BSW: New for CHV, BSW.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>FUSEDW17</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Spare</td>
</tr>
</tbody>
</table>
### FUSEWORD18

**Register Space:** MMIO: 0/2/0  
**Project:** CHV, BSW  
**Source:** PRM  
**Default Value:** 0x00000000  
**Size (in bits):** 32  
**Address:** 182178h

Fuse readout information. CHV, BSW: New for CHV, BSW.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
<th>Spare</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>FUSEDW18</strong></td>
<td>00000000h</td>
<td>RO</td>
<td>Yes</td>
</tr>
</tbody>
</table>

_{Fuse readout information. CHV, BSW: New for CHV, BSW._}
### FUSEWORD19 - FUSEWORD19

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>18217Ch</td>
</tr>
</tbody>
</table>

Fuse readout information. CHV, BSW: New for CHV, BSW.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>FUSEDW19</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Spare</td>
</tr>
</tbody>
</table>
## G3D Control Register

### G3DCTL - G3D Control Register

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:27</td>
<td><strong>G3D Spare</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Spare config bits sent to G3D.</td>
</tr>
<tr>
<td>26</td>
<td>RTN Mode</td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RTN mode. When set to one all GAM read returns are considered slow (no b2b to any source ID).</td>
</tr>
<tr>
<td>25</td>
<td>Return Ordering Queue Disable</td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ROQ Disable. When set to a '1', the G3D Return Ordering queue will be mostly disabled by forcing the Data queue to be considered full when it has one entry occupied.</td>
</tr>
<tr>
<td>24</td>
<td>Command Bandwidth Arb Mode</td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Command Bandwidth Arbitration mode. When programmed to a '1', reads will be allowed to go 1 cycle after writes as long as writes are not in write grant mode.</td>
</tr>
<tr>
<td>23:0</td>
<td><strong>RSVD</strong></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved.</td>
</tr>
</tbody>
</table>
## GAB Arbitration Programmable

<table>
<thead>
<tr>
<th><strong>GAB_AP - GAB Arbitration Programmable</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 040F0h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>DWord</strong></th>
<th><strong>Bit</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>Reserved</td>
</tr>
</tbody>
</table>
### GAB GAC GAM Idle

**GABGACGAMIDLE - GAB GAC GAM Idle**

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:26</td>
<td>Reserved</td>
<td>RO</td>
<td></td>
</tr>
<tr>
<td></td>
<td>25:16</td>
<td>GAM Idle Timeout</td>
<td>R/W</td>
<td><strong>Programming Notes</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>This mode is not supported. It must be 0 at all times.</td>
</tr>
<tr>
<td>15:10</td>
<td>Reserved</td>
<td></td>
<td>RO</td>
<td></td>
</tr>
<tr>
<td>9:0</td>
<td>Min GAB GAC Idle</td>
<td>R/W</td>
<td><strong>Programming Notes</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>This mode is not supported. It must be 0 at all times.</td>
<td></td>
</tr>
</tbody>
</table>
## GAB_LRA_0 - GAB LRA 0

**Register Space:** MMIO: 0/2/0  
**Project:** CHV, BSW  
**Source:** PRM  
**Default Value:** 0x1F100F00  
**Size (in bits):** 32  
**Address:** 04A70h

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:29</td>
<td><strong>Reserved</strong></td>
<td>000b</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td>28:24</td>
<td><strong>GAB LRA1 Max</strong></td>
<td>11111b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Maximum value of programmable LRA1.</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>23:21</td>
<td><strong>Reserved</strong></td>
<td>000b</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td>20:16</td>
<td><strong>GAB LRA1 Min</strong></td>
<td>10000b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Minimum value of programmable LRA1.</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>15:13</td>
<td><strong>Reserved</strong></td>
<td>000b</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td>12:8</td>
<td><strong>GAB LRA0 Max</strong></td>
<td>01111b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Maximum value of programmable LRA0.</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>7:5</td>
<td><strong>Reserved</strong></td>
<td>000b</td>
<td>RO</td>
</tr>
</tbody>
</table>
### GAB_LRA_0 - GAB LRA 0

<table>
<thead>
<tr>
<th>4:0</th>
<th>GABLRA0 Min</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Default Value: 00000b</td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

Minimum value of programmable LRA0.
# GAB LRA 1

## GAB_LRA_1 - GAB LRA 1

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:4</td>
<td><strong>Reserved</strong></td>
<td>00000000h</td>
<td>RO</td>
</tr>
<tr>
<td>3:2</td>
<td></td>
<td><strong>BLB</strong></td>
<td>00b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Which LRA should BLB use.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1:0</td>
<td></td>
<td><strong>BCS</strong></td>
<td>01b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Which LRA should BCS use.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Register Space:** MMIO: 0/2/0
- **Project:** CHV, BSW
- **Source:** PRM
- **Default Value:** 0x00000001
- **Size (in bits):** 32
- **Address:** 04A74h
# GAB unit Control Register

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>BlitterCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x000000BF</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>24000h</td>
</tr>
</tbody>
</table>

Default Value=FF0000BFh Trusted Type = 1

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

### ARB_GAC_GAM_REQCNTS0 - GAC_GAM Arbitration Counters Register 0

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>RenderCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Trusted Type:</td>
<td>1</td>
</tr>
<tr>
<td>Address:</td>
<td>043A8h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:22</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td>21:16</td>
<td>Number of GAC WR requests to be accumulated before applying the arbitration</td>
</tr>
<tr>
<td></td>
<td>15:14</td>
<td>Reserved</td>
</tr>
<tr>
<td>1</td>
<td>13:8</td>
<td>Number of GAC R requests to be accumulated before applying the arbitration</td>
</tr>
<tr>
<td>2</td>
<td>7:6</td>
<td>Reserved</td>
</tr>
<tr>
<td>3</td>
<td>5:0</td>
<td>Number of GAC RO requests to be accumulated before applying the arbitration</td>
</tr>
</tbody>
</table>
### GAC_GAM Arbitration Counters Register 1

**ARB_GAC_GAM_REQCNTS1 - GAC_GAM Arbitration Counters Register 1**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>RenderCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Trusted Type:</td>
<td>1</td>
</tr>
<tr>
<td>Address:</td>
<td>043ACh</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:22</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td>21:16</td>
<td>Number of GAC WR requests to be accumulated before applying the arbitration</td>
</tr>
<tr>
<td></td>
<td>15:14</td>
<td>Reserved</td>
</tr>
<tr>
<td>13:8</td>
<td></td>
<td>Number of GAC R requests to be accumulated before applying the arbitration</td>
</tr>
<tr>
<td>7:6</td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td>5:0</td>
<td></td>
<td>Number of GAC RO requests to be accumulated before applying the arbitration</td>
</tr>
</tbody>
</table>
### GAC_GAM R Arbitration Register 0

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:28</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td>27</td>
<td><strong>Priority for entry 1</strong></td>
</tr>
<tr>
<td></td>
<td>26:24</td>
<td><strong>Goto field for entry 1 when request vector is 11b</strong></td>
</tr>
<tr>
<td></td>
<td>23:21</td>
<td><strong>Goto field for entry 1 when request vector is 10b</strong></td>
</tr>
<tr>
<td></td>
<td>20:18</td>
<td><strong>Goto field for entry 1 when request vector is 01b</strong></td>
</tr>
<tr>
<td></td>
<td>17:15</td>
<td><strong>Goto field for entry 1 when request vector is 00b</strong></td>
</tr>
<tr>
<td></td>
<td>14:13</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td>12</td>
<td><strong>Priority for entry 0</strong></td>
</tr>
<tr>
<td></td>
<td>11:9</td>
<td><strong>Goto field for entry 0 when request vector is 11b</strong></td>
</tr>
<tr>
<td></td>
<td>8:6</td>
<td><strong>Goto field for entry 0 when request vector is 10b</strong></td>
</tr>
<tr>
<td></td>
<td>5:3</td>
<td><strong>Goto field for entry 0 when request vector is 01b</strong></td>
</tr>
<tr>
<td></td>
<td>2:0</td>
<td><strong>Goto field for entry 0 when request vector is 00b</strong></td>
</tr>
</tbody>
</table>
### GAC_GAM R Arbitration Register 1

<table>
<thead>
<tr>
<th>ARB_R_GAC_GAM1 - GAC_GAM R Arbitration Register 1</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> RenderCS</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Trusted Type:</strong> 1</td>
</tr>
<tr>
<td><strong>Address:</strong> 043E4h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:28</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td>27</td>
<td>Priority for entry 3</td>
</tr>
<tr>
<td></td>
<td>26:24</td>
<td>Goto field for entry 3 when request vector is 11b</td>
</tr>
<tr>
<td></td>
<td>23:21</td>
<td>Goto field for entry 3 when request vector is 10b</td>
</tr>
<tr>
<td></td>
<td>20:18</td>
<td>Goto field for entry 3 when request vector is 01b</td>
</tr>
<tr>
<td></td>
<td>17:15</td>
<td>Goto field for entry 3 when request vector is 00b</td>
</tr>
<tr>
<td></td>
<td>14:13</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>Priority for entry 2</td>
</tr>
<tr>
<td></td>
<td>11:9</td>
<td>Goto field for entry 2 when request vector is 11b</td>
</tr>
<tr>
<td></td>
<td>8:6</td>
<td>Goto field for entry 2 when request vector is 10b</td>
</tr>
<tr>
<td></td>
<td>5:3</td>
<td>Goto field for entry 2 when request vector is 01b</td>
</tr>
<tr>
<td></td>
<td>2:0</td>
<td>Goto field for entry 2 when request vector is 00b</td>
</tr>
</tbody>
</table>
## GAC_GAM R Arbitration Register 2

<table>
<thead>
<tr>
<th>ARB_R_GAC_GAM2 - GAC_GAM R Arbitration Register 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: RenderCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Access: R/W</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Trusted Type: 1</td>
</tr>
<tr>
<td>Address: 043E8h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:28</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td>27</td>
<td>Priority for entry 5</td>
</tr>
<tr>
<td></td>
<td>26:24</td>
<td>Goto field for entry 5 when request vector is 11b</td>
</tr>
<tr>
<td></td>
<td>23:21</td>
<td>Goto field for entry 5 when request vector is 10b</td>
</tr>
<tr>
<td></td>
<td>20:18</td>
<td>Goto field for entry 5 when request vector is 01b</td>
</tr>
<tr>
<td></td>
<td>17:15</td>
<td>Goto field for entry 5 when request vector is 00b</td>
</tr>
<tr>
<td></td>
<td>14:13</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>Priority for entry 4</td>
</tr>
<tr>
<td></td>
<td>11:9</td>
<td>Goto field for entry 4 when request vector is 11b</td>
</tr>
<tr>
<td></td>
<td>8:6</td>
<td>Goto field for entry 4 when request vector is 10b</td>
</tr>
<tr>
<td></td>
<td>5:3</td>
<td>Goto field for entry 4 when request vector is 01b</td>
</tr>
<tr>
<td></td>
<td>2:0</td>
<td>Goto field for entry 4 when request vector is 00b</td>
</tr>
</tbody>
</table>
### GAC_GAM R Arbitration Register 3

<table>
<thead>
<tr>
<th>ARB_R_GAC_GAM3 - GAC_GAM R Arbitration Register 3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> RenderCS</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Trusted Type:</strong> 1</td>
</tr>
</tbody>
</table>

**Address:** 043ECh

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:28</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td>27</td>
<td>Priority for entry 7</td>
</tr>
<tr>
<td></td>
<td>26:24</td>
<td>Goto field for entry 7 when request vector is 11b</td>
</tr>
<tr>
<td></td>
<td>23:21</td>
<td>Goto field for entry 7 when request vector is 10b</td>
</tr>
<tr>
<td></td>
<td>20:18</td>
<td>Goto field for entry 7 when request vector is 01b</td>
</tr>
<tr>
<td></td>
<td>17:15</td>
<td>Goto field for entry 7 when request vector is 00b</td>
</tr>
<tr>
<td></td>
<td>14:13</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>Priority for entry 6</td>
</tr>
<tr>
<td></td>
<td>11:9</td>
<td>Goto field for entry 6 when request vector is 11b</td>
</tr>
<tr>
<td></td>
<td>8:6</td>
<td>Goto field for entry 6 when request vector is 10b</td>
</tr>
<tr>
<td></td>
<td>5:3</td>
<td>Goto field for entry 6 when request vector is 01b</td>
</tr>
<tr>
<td></td>
<td>2:0</td>
<td>Goto field for entry 6 when request vector is 00b</td>
</tr>
</tbody>
</table>
GAC_GAM RO Arbitration Register 0

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:28</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td>27</td>
<td>Priority for entry 1</td>
</tr>
<tr>
<td></td>
<td>26:24</td>
<td>Goto field for entry 1 when request vector is 11b</td>
</tr>
<tr>
<td></td>
<td>23:21</td>
<td>Goto field for entry 1 when request vector is 10b</td>
</tr>
<tr>
<td></td>
<td>20:18</td>
<td>Goto field for entry 1 when request vector is 01b</td>
</tr>
<tr>
<td></td>
<td>17:15</td>
<td>Goto field for entry 1 when request vector is 00b</td>
</tr>
<tr>
<td></td>
<td>14:13</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>Priority for entry 01</td>
</tr>
<tr>
<td></td>
<td>11:9</td>
<td>Goto field for entry 01 when request vector is 11b</td>
</tr>
<tr>
<td></td>
<td>8:6</td>
<td>Goto field for entry 01 when request vector is 10b</td>
</tr>
<tr>
<td></td>
<td>5:3</td>
<td>Goto field for entry 01 when request vector is 01b</td>
</tr>
<tr>
<td></td>
<td>2:0</td>
<td>Goto field for entry 01 when request vector is 00b</td>
</tr>
</tbody>
</table>
# GAC_GAM RO Arbitration Register 1

<table>
<thead>
<tr>
<th>ARB_RO_GAC_GAM1 - GAC_GAM RO Arbitration Register 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: RenderCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Access: R/W</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Trusted Type: 1</td>
</tr>
<tr>
<td>Address: 043D4h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:28</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td>27</td>
<td>Priority for entry 3</td>
</tr>
<tr>
<td></td>
<td>26:24</td>
<td>Goto field for entry 3 when request vector is 11b</td>
</tr>
<tr>
<td></td>
<td>23:21</td>
<td>Goto field for entry 3 when request vector is 10b</td>
</tr>
<tr>
<td></td>
<td>20:18</td>
<td>Goto field for entry 3 when request vector is 01b</td>
</tr>
<tr>
<td></td>
<td>17:15</td>
<td>Goto field for entry 3 when request vector is 00b</td>
</tr>
<tr>
<td></td>
<td>14:13</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>Priority for entry 2</td>
</tr>
<tr>
<td></td>
<td>11:9</td>
<td>Goto field for entry 2 when request vector is 11b</td>
</tr>
<tr>
<td></td>
<td>8:6</td>
<td>Goto field for entry 2 when request vector is 10b</td>
</tr>
<tr>
<td></td>
<td>5:3</td>
<td>Goto field for entry 2 when request vector is 01b</td>
</tr>
<tr>
<td></td>
<td>2:0</td>
<td>Goto field for entry 2 when request vector is 00b</td>
</tr>
</tbody>
</table>
# GAC_GAM RO Arbitration Register 2

**ARB_RO_GAC_GAM2 - GAC_GAM RO Arbitration Register 2**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>RenderCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Trusted Type:</td>
<td>1</td>
</tr>
<tr>
<td>Address:</td>
<td>043D8h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:28</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td>27</td>
<td><strong>Priority for entry 5</strong></td>
</tr>
<tr>
<td></td>
<td>26:24</td>
<td><strong>Goto field for entry 5 when request vector is 11b</strong></td>
</tr>
<tr>
<td></td>
<td>23:21</td>
<td><strong>Goto field for entry 5 when request vector is 10b</strong></td>
</tr>
<tr>
<td></td>
<td>20:18</td>
<td><strong>Goto field for entry 5 when request vector is 01b</strong></td>
</tr>
<tr>
<td></td>
<td>17:15</td>
<td><strong>Goto field for entry 5 when request vector is 00b</strong></td>
</tr>
<tr>
<td></td>
<td>14:13</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td>12</td>
<td><strong>Priority for entry 4</strong></td>
</tr>
<tr>
<td></td>
<td>11:9</td>
<td><strong>Goto field for entry 4 when request vector is 11b</strong></td>
</tr>
<tr>
<td></td>
<td>8:6</td>
<td><strong>Goto field for entry 4 when request vector is 10b</strong></td>
</tr>
<tr>
<td></td>
<td>5:3</td>
<td><strong>Goto field for entry 4 when request vector is 01b</strong></td>
</tr>
<tr>
<td></td>
<td>2:0</td>
<td><strong>Goto field for entry 4 when request vector is 00b</strong></td>
</tr>
</tbody>
</table>
### GAC_GAM RO Arbitration Register 3

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>RenderCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Trusted Type:</td>
<td>1</td>
</tr>
<tr>
<td>Address:</td>
<td>043DCh</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:28</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td>27</td>
<td>Priority for entry 7</td>
</tr>
<tr>
<td></td>
<td>26:24</td>
<td>Goto field for entry 7 when request vector is 11b</td>
</tr>
<tr>
<td></td>
<td>23:21</td>
<td>Goto field for entry 7 when request vector is 10b</td>
</tr>
<tr>
<td></td>
<td>20:18</td>
<td>Goto field for entry 7 when request vector is 01b</td>
</tr>
<tr>
<td></td>
<td>17:15</td>
<td>Goto field for entry 7 when request vector is 00b</td>
</tr>
<tr>
<td></td>
<td>14:13</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>Priority for entry 6</td>
</tr>
<tr>
<td></td>
<td>11:9</td>
<td>Goto field for entry 6 when request vector is 11b</td>
</tr>
<tr>
<td></td>
<td>8:6</td>
<td>Goto field for entry 6 when request vector is 10b</td>
</tr>
<tr>
<td></td>
<td>5:3</td>
<td>Goto field for entry 6 when request vector is 01b</td>
</tr>
<tr>
<td></td>
<td>2:0</td>
<td>Goto field for entry 6 when request vector is 00b</td>
</tr>
</tbody>
</table>
## GAC_GAM WR Arbitration Register 0

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:28</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td>27</td>
<td>Priority for entry 1</td>
</tr>
<tr>
<td></td>
<td>26:24</td>
<td>Goto field for entry 1 when request vector is 11b</td>
</tr>
<tr>
<td></td>
<td>23:21</td>
<td>Goto field for entry 1 when request vector is 10b</td>
</tr>
<tr>
<td></td>
<td>20:18</td>
<td>Goto field for entry 1 when request vector is 01b</td>
</tr>
<tr>
<td></td>
<td>17:15</td>
<td>Goto field for entry 1 when request vector is 00b</td>
</tr>
<tr>
<td></td>
<td>14:13</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>Priority for entry 0</td>
</tr>
<tr>
<td></td>
<td>11:9</td>
<td>Goto field for entry 0 when request vector is 11b</td>
</tr>
<tr>
<td></td>
<td>8:6</td>
<td>Goto field for entry 0 when request vector is 10b</td>
</tr>
<tr>
<td></td>
<td>5:3</td>
<td>Goto field for entry 0 when request vector is 01b</td>
</tr>
<tr>
<td></td>
<td>2:0</td>
<td>Goto field for entry 0 when request vector is 00b</td>
</tr>
</tbody>
</table>
### GAC_GAM WR Arbitration Register 1

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>RenderCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Trusted Type:</td>
<td>1</td>
</tr>
<tr>
<td>Address:</td>
<td>043F4h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:28</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td>27</td>
<td>Priority for entry 3</td>
</tr>
<tr>
<td></td>
<td>26:24</td>
<td>Goto field for entry 3 when request vector is 11b</td>
</tr>
<tr>
<td></td>
<td>23:21</td>
<td>Goto field for entry 3 when request vector is 10b</td>
</tr>
<tr>
<td></td>
<td>20:18</td>
<td>Goto field for entry 3 when request vector is 01b</td>
</tr>
<tr>
<td></td>
<td>17:15</td>
<td>Goto field for entry 3 when request vector is 00b</td>
</tr>
<tr>
<td>14:13</td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>Priority for entry 2</td>
</tr>
<tr>
<td>11:9</td>
<td></td>
<td>Goto field for entry 2 when request vector is 11b</td>
</tr>
<tr>
<td>8:6</td>
<td></td>
<td>Goto field for entry 2 when request vector is 10b</td>
</tr>
<tr>
<td>5:3</td>
<td></td>
<td>Goto field for entry 2 when request vector is 01b</td>
</tr>
<tr>
<td>2:0</td>
<td></td>
<td>Goto field for entry 2 when request vector is 00b</td>
</tr>
</tbody>
</table>
## GAC_GAM WR Arbitration Register 2

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

Register Space: MMIO: 0/2/0  
Project: CHV, BSW  
Source: RenderCS  
Default Value: 0x00000000  
Access: R/W  
Size (in bits): 32  
Trusted Type: 1  
Address: 043F8h
GAC_GAM WR Arbitration Register 3

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:28</td>
<td>Reserved</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td>27</td>
<td>Priority for entry 7</td>
<td>Priority for entry 7</td>
</tr>
<tr>
<td></td>
<td>26:24</td>
<td>Goto field for entry 7 when request vector is 11b</td>
<td>Goto field for entry 7 when request vector is 11b</td>
</tr>
<tr>
<td></td>
<td>23:21</td>
<td>Goto field for entry 7 when request vector is 10b</td>
<td>Goto field for entry 7 when request vector is 10b</td>
</tr>
<tr>
<td></td>
<td>20:18</td>
<td>Goto field for entry 7 when request vector is 01b</td>
<td>Goto field for entry 7 when request vector is 01b</td>
</tr>
<tr>
<td></td>
<td>17:15</td>
<td>Goto field for entry 7 when request vector is 00b</td>
<td>Goto field for entry 7 when request vector is 00b</td>
</tr>
<tr>
<td></td>
<td>14:13</td>
<td>Reserved</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>Priority for entry 6</td>
<td>Priority for entry 6</td>
</tr>
<tr>
<td></td>
<td>11:9</td>
<td>Goto field for entry 6 when request vector is 11b</td>
<td>Goto field for entry 6 when request vector is 11b</td>
</tr>
<tr>
<td></td>
<td>8:6</td>
<td>Goto field for entry 6 when request vector is 10b</td>
<td>Goto field for entry 6 when request vector is 10b</td>
</tr>
<tr>
<td></td>
<td>5:3</td>
<td>Goto field for entry 6 when request vector is 01b</td>
<td>Goto field for entry 6 when request vector is 01b</td>
</tr>
<tr>
<td></td>
<td>2:0</td>
<td>Goto field for entry 6 when request vector is 00b</td>
<td>Goto field for entry 6 when request vector is 00b</td>
</tr>
</tbody>
</table>
# GAM and SA Communication Register

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td><strong>Mask Bits</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Mask Bits act as Write Enables for the bits[15:0] of this register.</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td><strong>GAM and SA Communication Register 15</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For Future Use.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This bit is self clear.</td>
</tr>
<tr>
<td>14</td>
<td></td>
<td><strong>GAM and SA Communication Register 14</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For Future Use.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This bit is self clear.</td>
</tr>
<tr>
<td>13</td>
<td></td>
<td><strong>GAM and SA Communication Register 13</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For Future Use.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This bit is self clear.</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td><strong>GAM and SA Communication Register 12</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For Future Use.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This bit is self clear.</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td><strong>GAM and SA Communication Register 11</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For Future Use.</td>
</tr>
<tr>
<td></td>
<td>GAMSACOMREG - GAM and SA Communication Register</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td><strong>GAM and SA Communication Register 10</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default Value: 0b</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td>For Future Use.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>This bit is self clear.</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td><strong>GAM and SA Communication Register 9</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default Value: 0b</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td>For Future Use.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>This bit is self clear.</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td><strong>GAM and SA Communication Register 8</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default Value: 0b</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td>For Future Use.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>This bit is self clear.</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td><strong>GAM and SA Communication Register 7</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default Value: 0b</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td>For Future Use.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>This bit is self clear.</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td><strong>GAM and SA Communication Register 6</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default Value: 0b</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td>For Future Use.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>This bit is self clear.</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td><strong>GAM and SA Communication Register 5</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default Value: 0b</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td>For Future Use.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>This bit is self clear.</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td><strong>GAM and SA Communication Register 4</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default Value: 0b</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td>For Future Use.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>This bit is self clear.</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td><strong>GAM and SA Communication Register 3</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default Value: 0b</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td>For Future Use.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>This bit is self clear.</td>
<td></td>
</tr>
</tbody>
</table>
### GAMSACOMREG - GAM and SA Communication Register

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td><strong>GAM and SA Communication Register 2</strong></td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>Bit2 - Root Table Address Update Request.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>This bit is self clear.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td><strong>GAM and SA Communication Register 1</strong></td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>Bit1 - Queued Descriptor Request.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>This bit is self clear.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td><strong>GAM and SA Communication Register 0</strong></td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>Bit0 - Context Cache Invalidator Request.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>This bit is self clear.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Gam Fub Done1 Lookup Register

<table>
<thead>
<tr>
<th>DONE1_REG - Gam Fub Done1 Lookup Register</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 0407Ch</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>Gam Fub Done1 Lookup Reg</td>
</tr>
</tbody>
</table>

Default Value: 00000000h
Access: RO
GAM Done1 signals.
Gam Fub Done Lookup Register

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>Gam Fub Done Lookup Reg</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>31</td>
<td>CSV Credit Fifo is empty.</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>CVS TLB does not have any cycles.</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>Z Credit fifo is empty.</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>ZTLB does not have any cycles.</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>RCC Credit Fifo is empty.</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>RCC TLB does not have any cycles.</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>L3 Credit fifo is empty.</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>L3 TLB does not have any cycles.</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>VLF Credit fifo is empty.</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>VLF TLB does not have any cycles.</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>CASC Credit fifo empty.</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>CASC TLB does not have any cycles.</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>Miss Fub Done.</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>Read Stream Done.</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Read Steam Fifo is empty.</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Recycle Fifo in rstrm is empty.</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>TLB Pend Done.</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>TLB Pend PQ Array is done.</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>TLB pend PB Array is done.</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Read route fub is done.</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Gafm Data fifo is empty.</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>GAP data fifo is empty.</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>GAC data fifo is empty.</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Wrdp is done with all the cycles.</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Wrdp RID fifo is empty.</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>No hold from midarb to RTSTRM.</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>No hold from TLBPEND to MIDARB.</td>
<td></td>
</tr>
</tbody>
</table>
**DONE_REG - Gam Fub Done Lookup Register**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Tied to &quot;1&quot; - to be defined.</td>
</tr>
<tr>
<td>2</td>
<td>Fence FSM are IDLE.</td>
</tr>
<tr>
<td>1</td>
<td>Non PD Load Done.</td>
</tr>
<tr>
<td>0</td>
<td>Tied to &quot;1&quot; - to be defined.</td>
</tr>
</tbody>
</table>
## GAM Put Delay

### GAM.PUT.DLY - GAM Put Delay

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0401Ch</td>
</tr>
</tbody>
</table>

Number of clocks to wait between puts

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>GAM.PUT.DELAY</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>
### GAMT_DONE Register

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>GAMT_DONE Register</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td>31:</td>
<td>vebxtlb_all_done_f</td>
</tr>
<tr>
<td></td>
<td>30:</td>
<td>cvstlb_all_done_f</td>
</tr>
<tr>
<td></td>
<td>29:</td>
<td>ztlb_all_done_f</td>
</tr>
<tr>
<td></td>
<td>28:</td>
<td>l3tlb_all_done_f</td>
</tr>
<tr>
<td></td>
<td>27:</td>
<td>rcctlb_all_done_f</td>
</tr>
<tr>
<td></td>
<td>26:</td>
<td>mfxtlb_all_done_f</td>
</tr>
<tr>
<td></td>
<td>25:</td>
<td>vlftlb_all_done_f</td>
</tr>
<tr>
<td></td>
<td>24:</td>
<td>bwgtlb_all_done_f</td>
</tr>
<tr>
<td></td>
<td>23:</td>
<td>gamwrrb_all_done_f</td>
</tr>
<tr>
<td></td>
<td>22:</td>
<td>mfxs1tlb_all_done_f</td>
</tr>
<tr>
<td></td>
<td>21:</td>
<td>vlfsl1tlb_all_done_f</td>
</tr>
<tr>
<td></td>
<td>20:</td>
<td>bwgtlb_fifo_empty</td>
</tr>
<tr>
<td></td>
<td>19:</td>
<td>l3tlb_fifo_empty</td>
</tr>
<tr>
<td></td>
<td>18:</td>
<td>ztlb_fifo_empty</td>
</tr>
<tr>
<td></td>
<td>17:</td>
<td>rcctlb_fifo_empty</td>
</tr>
<tr>
<td></td>
<td>16:</td>
<td>cvstlb_fifo_empty</td>
</tr>
<tr>
<td></td>
<td>15:</td>
<td>vebxtlb_fifo_empty</td>
</tr>
<tr>
<td></td>
<td>14:</td>
<td>mfxtlb_fifo_empty</td>
</tr>
<tr>
<td></td>
<td>13:</td>
<td>mfxs1tlb_fifo_empty</td>
</tr>
<tr>
<td></td>
<td>12:</td>
<td>vlfsl1tlb_fifo_empty</td>
</tr>
<tr>
<td></td>
<td>11:</td>
<td>vlftlb_fifo_empty</td>
</tr>
<tr>
<td></td>
<td>10:</td>
<td>wrdp_gafm_fifo_empty</td>
</tr>
<tr>
<td></td>
<td>9:</td>
<td>wrdp_gap_fifo_empty</td>
</tr>
<tr>
<td></td>
<td>8:</td>
<td>wrdp_gacfg_fifo_empty</td>
</tr>
<tr>
<td></td>
<td>7:</td>
<td>wrdp_cs_fifo_empty</td>
</tr>
<tr>
<td></td>
<td>6:</td>
<td>wrdp_vecs_fifo_empty</td>
</tr>
<tr>
<td></td>
<td>5:</td>
<td>wrdp_oacs_fifo_empty</td>
</tr>
<tr>
<td></td>
<td>4:</td>
<td>wrdp_gacv_fifo_empty</td>
</tr>
</tbody>
</table>
| GAMT_DONE - GAMT_DONE Register | 3: Tied to 1  
|                               | 2: Tied to 1  
|                               | 1: Tied to 1  
|                               | 0: Tied to 1  |
## GAMT_ECO_REG_RO_IA

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>04AB4h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>GAMTECO_REG_RO_IA</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
</tbody>
</table>

This register is for ECO usage. RO register with IA Access Type on DEV reset.
## GAMT_ECO_REG_RW_IA

<table>
<thead>
<tr>
<th>GAMT_ECO_REG_RW_IA - GAMT_ECO_REG_RW_IA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW, GT2:B</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x0000AB1B</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 04AB0h</td>
</tr>
</tbody>
</table>

**Programmable Request Count - VEBX and BLT**

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>GAMTECO_REG_RW_IA</strong></td>
</tr>
</tbody>
</table>

| Default Value: | 0000AB1Bh |
| Access:        | R/W       |

Bit[31:16] = Reserved.
Bit[15:8] = Number of max outstanding cycles (Misses and Hits not present) that can be allowed to potentially fault = 171.
Bit[5:0] = Number of max outstanding misses that can be allowed to potentially fault = 27.
# GAMT Arbiter Mode Control

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td><strong>Mask Bits</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td><strong>GAMT Arbiter Mode Control 15</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For Future Use</td>
</tr>
<tr>
<td>14</td>
<td></td>
<td><strong>GAMT Arbiter Mode Control 14</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 - Cache the TLB even if there is a FAULT in GAMW read return.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 - Don’t Cache the TLB if there is a fault in GAMW return.</td>
</tr>
<tr>
<td>13</td>
<td></td>
<td><strong>GAMT Arbiter Mode Control 13</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 - VEBXTLB clock gate enabled.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 - VEBXTLB clock gate disabled.</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td><strong>GAMT Arbiter Mode Control 12</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 - MFXSL1TLB clock gate enabled.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 - MFXSL1TLB clock gate disabled.</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td><strong>GAMT Arbiter Mode Control 11</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 - VLFSL1TLB clock gate enabled.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 - VLFSL1TLB clock gate disabled.</td>
</tr>
<tr>
<td>Command Reference: Registers</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>GAMTARBMODE - GAMT Arbiter Mode Control</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>10</th>
<th>GAMT Arbiter Mode Control 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Value:</td>
<td>0b</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>0 - gamwrrb clock gate enabled.</td>
<td>1 - gamwrrb clock gate disabled.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>9</th>
<th>GAMT Arbiter Mode Control 9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Value:</td>
<td>0b</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>0 - BWGTLB clock gate enabled.</td>
<td>1 - BWGTLB clock gate disabled.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>8</th>
<th>GAMT Arbiter Mode Control 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Value:</td>
<td>0b</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>0 - VLFTLB clock gate enabled.</td>
<td>1 - VLFTLB clock gate disabled.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>7</th>
<th>GAMT Arbiter Mode Control 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Value:</td>
<td>0b</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>0 - MFXTLB clock gate enabled.</td>
<td>1 - MFXTLB clock gate disabled.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>6</th>
<th>GAMT Arbiter Mode Control 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Value:</td>
<td>0b</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>0 - RCCTLB clock gate enabled.</td>
<td>1 - RCCTLB clock gate disabled.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>5</th>
<th>GAMT Arbiter Mode Control 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Value:</td>
<td>0b</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>0 - L3TLB clock gate enabled.</td>
<td>1 - L3TLB clock gate disabled.</td>
</tr>
<tr>
<td>bit[5] needs to be set as a work-around due to recent gacb bug.</td>
<td></td>
</tr>
<tr>
<td>To update bit 5, a value of 0x00200020 needs to be written.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>4</th>
<th>GAMT Arbiter Mode Control 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Value:</td>
<td>0b</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>0 - ZTLB clock gate enabled.</td>
<td>1 - ZTLB clock gate disabled.</td>
</tr>
</tbody>
</table>
## GAMTARBMODE - GAMT Arbiter Mode Control

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>GAMT Arbiter Mode Control 3</td>
<td>0b</td>
<td>R/W</td>
<td>0 - CVS clock gate enabled. 1 - CVS clock gate disabled.</td>
</tr>
<tr>
<td>2</td>
<td>GAMT Arbiter Mode Control 2</td>
<td>0b</td>
<td>R/W</td>
<td>0 - No reg_hdc_inval_ack_force - take the value from client. 1 - reg_hdc_inval_ack_force - force value to 1 - disregard client value.</td>
</tr>
<tr>
<td>1</td>
<td>GAMT Arbiter Mode Control 1</td>
<td>0b</td>
<td>R/W</td>
<td>Bit [1]: Address Swizzling for Tiled Surfaces. This register location is updated via GFX Driver prior to enabling DRAM accesses. Driver needs to obtain the need for memory address swizzling via DRAM configuration registers and set the following bits (in Display Engine and Render/Media access streams). 0: No address Swizzling. 1: Address bit[1] needs to be swizzled for tiled surfaces.</td>
</tr>
<tr>
<td>0</td>
<td>GAMT Arbiter Mode Control 0</td>
<td>0b</td>
<td>R/W</td>
<td>Bit[0]: GAM to Bypass GTT Translation. GAM to Bypass GTT Translation and pass logical addresses through with 0's padded on the MSBs to form the Physical Address.</td>
</tr>
</tbody>
</table>
## GAMW_ECO_BUS_RO_IA

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0408Ch</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>GAMWECO_BUS_RO_IA</td>
</tr>
</tbody>
</table>

- **Default Value:** 00000000h
- **Access:** RO

This register is for ECO usage. RO register with IA Access Type on BUS reset.
# GAMW_ECO_BUS_RW_IA

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>04084h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>GAMWECO_BUS_RW_IA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

This register is for ECO usage. RW register with IA Access Type on BUS reset.
# GAMW_ECO_DEV_RO_IA

<table>
<thead>
<tr>
<th><strong>GAMW_ECO_DEV_RO_IA - GAMW_ECO_DEV_RO_IA</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong></td>
</tr>
<tr>
<td><strong>Project:</strong></td>
</tr>
<tr>
<td><strong>Source:</strong></td>
</tr>
<tr>
<td><strong>Default Value:</strong></td>
</tr>
<tr>
<td><strong>Size (in bits):</strong></td>
</tr>
<tr>
<td><strong>Address:</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>DWord</strong></th>
<th><strong>Bit</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>GAMWECO_DEV_RO_IA</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> RO</td>
</tr>
</tbody>
</table>

This register is for ECO usage. RO register with IA Access Type on DEV reset.
## GAMW_ECO_DEV_RW_IA

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>04080h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>GAMWECO_DEV_RW_IA</td>
</tr>
</tbody>
</table>

- **Default Value:** 00000000h
- **Access:** R/W
# GAMW Power Context Save

## PWRCTXSAVE - GAMW Power Context Save

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td><strong>Mask Bits</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Mask Bits act as Write Enables for the bits[15:0] of this register.</td>
</tr>
<tr>
<td>15</td>
<td>Extra Bits15</td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Extra Bits for future use.</td>
</tr>
<tr>
<td>14</td>
<td>Extra Bits14</td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Extra Bits for future use.</td>
</tr>
<tr>
<td>13</td>
<td>Extra Bits13</td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Extra Bits for future use.</td>
</tr>
<tr>
<td>12</td>
<td>Extra Bits12</td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Extra Bits for future use.</td>
</tr>
<tr>
<td>11</td>
<td>Extra Bits11</td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Extra Bits for future use.</td>
</tr>
<tr>
<td>10</td>
<td>Extra Bits10</td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Extra Bits for future use.</td>
</tr>
</tbody>
</table>
### PWRCTXSAVE - GAMW Power Context Save

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>Power Context Save Request</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td>8:0</td>
<td>Power Context Save Quad Word Credits</td>
<td>000000000b</td>
<td>R/W</td>
</tr>
</tbody>
</table>

**Power Context Save Request**

- **Bit[9]**
  - Power Context Save Request
  - 1'b0: Power context save is not being requested (default).
  - 1'b1: Power context save is being requested.
  - Unit needs to self-clear this bit upon sampling.
  - This bit is self clear.

**Power Context Save Quad Word Credits**

- **Bits[8:0]**
  - QWord Credits for Power Context Save Request
  - An initial length packet is required per power context save session, but that packet does not consume a credit. See protocol description for more details.
  - Minimum Credits = 1: Unit may send 1 QWord pair.
  - Maximum Credits = 511: Unit may send 511 QWord pairs.
  - A QWord pair is defined as a 32-bit register address and the corresponding 32-bits of register data.
  - Only valid with PWRCTX_SAVE_REQ (Bit9).
GARB Messaging Register for Boot Controller

<table>
<thead>
<tr>
<th>MSG_GARB_MBC - GARB Messaging Register for Boot Controller</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 16</td>
</tr>
<tr>
<td>Address: 0801Ch</td>
</tr>
</tbody>
</table>

Message registers have bit-wise masking applied for writes. The register consists of 16 bits of data in [15:0], and 16 bits of corresponding masks in [31:16]. In order to set bit0, for example, the data would be 0x0001_0001. In order to clear bit0, for example, the data would be 0x0001_0000. Note that mask bit is the data bit offset + 16. Message registers are protected from non-GT writes via the Message Channel.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>15:7</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>Fuse Download Done Indication</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fuse Download Done Indication 1'b0 : Fuse download is not complete yet &lt;default&gt; 1'b1 : Fuse download is complete gpmunit self-clears this bit upon sampling.</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>Boot Fetch Complete Indication</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Boot Fetch Complete Indication 1'b0 : Boot Fetch is not complete yet &lt;default&gt; 1'b1 : Boot Fetch is complete gpmunit self-clears this bit upon sampling.</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>IDI Block Status</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IDI Block Status 1'b0 : IDI interface is not blocked &lt;default&gt; 1'b1 : IDI interface is blocked</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>IDI Awake Status</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IDI Awake Status 1'b0 : IDI interface is not ready &lt;default&gt; 1'b1 : IDI interface is ready</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>Credit Active Status</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Credit Active Status 1'b0 : Send credit active deassert Event Bus Message on transition from 1'b1 =&gt; 1'b0 &lt;default&gt; 1'b1 : Send credit active assert Event Bus Message on transition from 1'b0 =&gt; 1'b1</td>
</tr>
</tbody>
</table>
### MSG_GARB_MBC - GARB Messaging Register for Boot Controller

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><strong>Global Arbitration Request</strong></td>
<td>R/W</td>
<td>1'b0 : No request &lt;default&gt; 1'b1 : Request for arbitration Full handshake requiring ack</td>
</tr>
<tr>
<td>0</td>
<td><strong>Busy Indication</strong></td>
<td>R/W</td>
<td>1'b0 : Idle &lt;default&gt; 1'b1 : Busy Full handshake requiring ack</td>
</tr>
</tbody>
</table>
### GARB Messaging Register for Clocking Unit

<table>
<thead>
<tr>
<th>MSG_GARB_GCP - GARB Messaging Register for Clocking Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 16</td>
</tr>
<tr>
<td><strong>Address:</strong> 08024h</td>
</tr>
</tbody>
</table>

Message registers have bit-wise masking applied for writes. The register consists of 16 bits of data in [15:0], and 16 bits of corresponding masks in [31:16]. In order to set bit0, for example, the data would be 0x0001_0001. In order to clear bit0, for example, the data would be 0x0001_0000. Note that mask bit is the data bit offset + 16. Message registers are protected from non-GT writes via the Message Channel.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>15:3</td>
<td><strong>Reserved</strong></td>
<td>RO</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td><strong>GCP Request to send FLR Complete Message to SA via GAM</strong></td>
<td>R/W</td>
<td>GCP Request to send FLR Complete Message to SA via GAM&lt;br&gt;1'b1 : Send cycle on GA* Interface to address &lt; address &gt; with data &lt; data &gt;&lt;br&gt;gpm will self-clear the request once it completes it. MBC needs to self-clear the acknowledgement once it sees it. GPM indicates a write cycle is complete once it puts it on the interface. GPM indicates a read cycle is complete once the read-return data comes back.&lt;br&gt;gpmunit self-clears this bit upon sampling.</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td><strong>Global Arbitration Request</strong></td>
<td>R/W</td>
<td>Global Arbitration Request&lt;br&gt;1'b0 : No request &lt;default&gt;&lt;br&gt;1'b1 : Request for arbitration&lt;br&gt;Full handshake requiring ack</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td><strong>Busy Indication</strong></td>
<td>R/W</td>
<td>Busy Indication&lt;br&gt;1'b0 : Idle &lt;default&gt;&lt;br&gt;1'b1 : Busy&lt;br&gt;Full handshake requiring ack</td>
</tr>
</tbody>
</table>
Gather Constants Not Consumed By RCS

<table>
<thead>
<tr>
<th>GATHER_CONST_PRODUCE_COUNT - Gather Constants Not Consumed By RCS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> RenderCS</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Trusted Type:</strong> 1</td>
</tr>
<tr>
<td><strong>Address:</strong> 0248Ch</td>
</tr>
</tbody>
</table>

This register keeps track of the outstanding Gather Constants produced by RS which are not yet consumed by Render Command Streamer. This register is part of the render context save and restore. This register should not be programmed by SW.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>Gather Constants Produce Count</td>
</tr>
</tbody>
</table>

This register keeps track of the outstanding Gather Constants produced by RS which are not yet consumed by Render Command Streamer. This register is part of the render context save and restore.
## WR_DROP_MODE - GDR Per Client Write Drop Enables

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>GDR Per Client Write Drop Enables</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
</tbody>
</table>

- 31 RSVD: Future use.
- 30 MBC write drop disable (0) or enable (1).
- 29 CS write drop disable (0) or enable (1).
- 28 SOL write drop disable (0) or enable (1).
- 27 RS write drop disable (0) or enable (1).
- 26 RCC write drop disable (0) or enable (1).
- 25 MSC write drop disable (0) or enable (1).
- 24 All L3 clients write drop disable (0) or enable (1).
- 23 STC write drop disable (0) or enable (1).
- 22 HIZ write drop disable (0) or enable (1).
- 21 RCZ write drop disable (0) or enable (1).
- 20 GAFS write drop disable (0) or enable (1).
- 19 GPM write drop disable (0) or enable (1).
- 18 GCP write drop disable (0) or enable (1).
- 17 VCS write drop disable (0) or enable (1).
- 16 BSP write drop disable (0) or enable (1).
- 15 VCR write drop disable (0) or enable (1).
- 14 VMX_RS write drop disable (0) or enable (1).
- 13 VMX_BS write drop disable (0) or enable (1).
- 12 VMX_RA write drop disable (0) or enable (1).
- 11 VMX_VDS write drop disable (0) or enable (1).
- 10 VLF_RS write drop disable (0) or enable (1).
- 9 VLF_FW write drop disable (0) or enable (1).
- 8 VECs write drop disable (0) or enable (1).
- 7 VEO write drop disable (0) or enable (1).
<table>
<thead>
<tr>
<th>WR DROP MODE - GDR Per Client Write Drop Enables</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 uC (DMA) write drop disable (0) or enable (1).</td>
</tr>
<tr>
<td>4 BCS write drop disable (0) or enable (1).</td>
</tr>
<tr>
<td>3 BLB write drop disable (0) or enable (1).</td>
</tr>
<tr>
<td>2 W_BSP write drop disable (0) or enable (1).</td>
</tr>
<tr>
<td>1 W_VMX_RS write drop disable (0) or enable (1).</td>
</tr>
<tr>
<td>0 W_VMX_BS write drop disable (0) or enable (1).</td>
</tr>
</tbody>
</table>
## GDR Write Drop

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>GDR_WRITE_DROP</strong></td>
<td>00000000h</td>
<td>R/W</td>
</tr>
</tbody>
</table>

- **Register Space:** MMIO: 0/2/0
- **Project:** CHV, BSW
- **Source:** PRM
- **Default Value:** 0x00000000
- **Size (in bits):** 32
- **Address:** 04020h
### General Purpose Power Management Performance Idle Hysteresis

**GPMPIHYST - General Purpose Power Management Performance Idle Hysteresis**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0A070h-0A073h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td>23:0</td>
<td></td>
<td><strong>Performance Idle Hysteresis Direction</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
</tbody>
</table>

Idle intervals must be longer than this value to be considered idle.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0 usec</td>
</tr>
<tr>
<td>1</td>
<td>1.28 usec</td>
</tr>
<tr>
<td>2</td>
<td>2.56 usec</td>
</tr>
<tr>
<td>3</td>
<td>3.84 usec</td>
</tr>
<tr>
<td>FF FFFF</td>
<td>21.474 sec</td>
</tr>
</tbody>
</table>

FYI: 0 means disabled.
### GFX_FLSH_CNT - GFX_FLSH_CNT

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>101008h</td>
</tr>
</tbody>
</table>

**Used to flush Gunit TLB**

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:1</td>
<td>RESERVED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td>0</td>
<td>GfxFlshCntl</td>
<td>Access type of this register is WO.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: WO</td>
</tr>
</tbody>
</table>

A write to this bit flushes the Gfx TLB in GUNIT. The data associated with the write is discarded and a read return all 0s.
# GFX Arbiter Client Priority Control

<table>
<thead>
<tr>
<th>GFX_PRIO_CTRL - GFX Arbiter Client Priority Control</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x880A2D10</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 04A00h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:27</td>
<td><strong>Read Rstrm Max Reject</strong></td>
<td>10001b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>26:21</td>
<td><strong>Extra Bits</strong></td>
<td>000000b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>20:18</td>
<td><strong>sol_gam_priority</strong></td>
<td>010b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>17:15</td>
<td><strong>veo_gam_priority</strong></td>
<td>100b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>14:12</td>
<td><strong>vfw_gam_priority</strong></td>
<td>010b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>11:9</td>
<td><strong>gapc_gam_c_priority</strong></td>
<td>110b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>8:6</td>
<td><strong>gapc_gam_z_priority</strong></td>
<td>100b</td>
<td>R/W</td>
</tr>
</tbody>
</table>

Client Priority Control Bits - Lowest Bit [18] is NOT Used.


Client Priority Control Bits - Lowest Bit [12] is NOT Used.

Client Priority Control Bits - Lowest Bit [9] is NOT Used.

## GFX_PRIO_CTRL - GFX Arbiter Client Priority Control

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>5:3</td>
<td>gapc_gam_i3_priority</td>
<td>010b</td>
<td>R/W</td>
<td>Client Priority Control Bits - Lowest Bit [3] is NOT Used.</td>
</tr>
<tr>
<td>2:0</td>
<td>csrsvf_gam_priority</td>
<td>000b</td>
<td>R/W</td>
<td>Client Priority Control Bits - Lowest Bit [0] is NOT Used.</td>
</tr>
</tbody>
</table>
### GFX Context Element Descriptor (High Part)

<table>
<thead>
<tr>
<th>GFX_CTX_EDR_H - GFX Context Element Descriptor (High Part)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 04404h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>GFX Context Element Descriptor (High Part)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td>Bit[63:32] - Context ID:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Context identification number assigned to separate this context from others. Context IDs need to be recycled in such a way that there cannot be two active contexts with the same ID. This is a unique identification number by which a context is identified and referenced.</td>
<td></td>
</tr>
</tbody>
</table>
### GFX Context Element Descriptor (Low Part)

<table>
<thead>
<tr>
<th>GFX_CTX_EDR_L - GFX Context Element Descriptor (Low Part)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000009</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 04400h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>GFX Context Element Descriptor (Low Part)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000009h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

- Bit[8] - Privileged Context / GGTT vs PPGTT mode: Differs in legacy vs advanced context modes:
  - In Legacy Context: Defines the page tables to be used. This is how page walker comes to know PPGTT vs GGTT selection for the entire context.
  - 0: Use Global GTT.
  - 1: Use Per-Process GTT.
  - In Advanced Context: Defines the privilege level for the context.
  - 0: User mode context.
  - 1: Supervisor mode context.
- Bit[5] - Deeper IA coherency Support:
  - In Advanced Context: Defines the level of IA coherency.
  - 0: IA coherency is provided at LLC level for all streams of GPU (i.e. Gen7.5 like mode).
  - 1: IA coherency is provided at L3 level for EU data accesses of GPU.
- Bit[4] - A and D Support / 32 and 64b Address Support: Differs in legacy vs advanced context modes:
  - In Legacy Context: Defines 32b vs 64b (48b canonical) addressing format:
    - 0: 32b addressing format.
    - 1: 64b (48b canonical) addressing format.
  - In Advanced Context: Defines A and D bit support:
    - 0: A and D bit management in page tables is NOT supported.
    - 1: A and D bit management in page tables is supported.
- Bit[3] - Context Type: Legacy vs Advanced:
  - Defines the context type
    - 0: Advanced Context: Defines the rest of the advanced capabilities (i.e. OS page table support, fault models). Note that advanced context is not bounded to GPGPU.
    - 1: Legacy Context: Defines the context as legacy mode which is similar to prior generations of Gen8.
  - Note that: Bits [8:4] differs in functions when legacy vs advanced context modes are selected.
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Bit[1] - Scheduling Mode:</td>
</tr>
<tr>
<td>1: Indicates excelist mode of scheduling.</td>
</tr>
<tr>
<td>0: Indicates Ring Buffer mode of scheduling.</td>
</tr>
<tr>
<td>Bit[0] - Valid: Indicates that element descriptor is valid. If GAM is programmed with an invalid descriptor, it continues but flags an error.</td>
</tr>
</tbody>
</table>
### GFX_CTX_EDR_L - GFX Context Element Descriptor (Low Part)

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000009</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>04400h</td>
</tr>
</tbody>
</table>

**DWord** | **Bit** | **Description** |
---|---|---|
| 0 | 31:0 | GFX Context Element Descriptor |
|   |   | Default Value: 00000009h |
|   |   | Access: R/W |

- **Bit[31:12] - LRCA:** Command Streamer Only.
- **Bit[8] - Privileged Context / GGTT vs PPGTT mode:** Differs in legacy vs advanced context modes:
  - In Legacy Context: Defines the page tables to be used. This is how page walker comes to know PPGTT vs GGTT selection for the entire context.
    - 0: Use Global GTT.
    - 1: Use Per-Process GTT.
  - In Advanced Context: Defines the privilege level for the context.
    - 0: User mode context.
    - 1: Supervisor mode context.

- **Bit[5] - Deeper IA coherency Support:**
  - In Advanced Context: Defines the level of IA coherency.
    - 0: IA coherency is provided at LLC level for all streams of GPU (i.e. Gen7.5 like mode).
    - 1: IA coherency is provided at L3 level for EU data accesses of GPU.
  - **Bit[4] - A and D Support / 32 and 64b Address Support:** Differs in legacy vs advanced context modes:
    - In Legacy Context: Defines 32b vs 64b (48b canonical) addressing format:
      - 0: 32b addressing format.
      - 1: 64b (48b canonical) addressing format.
    - In Advanced Context: Defines A and D bit support:
      - 0: A and D bit management in page tables is NOT supported.
      - 1: A and D bit management in page tables is supported.
  - **Bit[3] - Context Type: Legacy vs Advanced:**
    - Defines the context type
    - 0: Advanced Context: Defines the rest of the advanced capabilities (i.e. OS page table support, fault models). Note that advanced context is not bounded to GPGPU.
    - 1: Legacy Context: Defines the context as legacy mode which is similar to prior generations of Gen8.

Note that: Bits [8:4] differs in functions when legacy vs advanced context modes are selected.
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Bit[1]</td>
<td>Scheduling Mode:</td>
</tr>
<tr>
<td></td>
<td>1: Indicates execlist mode of scheduling.</td>
</tr>
<tr>
<td></td>
<td>0: Indicates Ring Buffer mode of scheduling.</td>
</tr>
<tr>
<td>Bit[0]</td>
<td>Valid: Indicates that element descriptor is valid. If GAM is programmed with an invalid descriptor, it continues but flags an error.</td>
</tr>
</tbody>
</table>
# GFX PDP0/PML4/PASID Descriptor (High Part)

<table>
<thead>
<tr>
<th>GFX_CTX_PDP0_H - GFX PDP0/PML4/PASID Descriptor (High Part)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 0440Ch</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>GFX PDP0/PML4/PASID Descriptor (High Part)</td>
</tr>
</tbody>
</table>

- **Default Value:** 00000000h
- **Access:** R/W

**PDP0/PML4/PASID:**
This register can contain three values which depend on the element descriptor definition.

- **PASID[19:0]:** Populated in the first 20bits of the register and selected when Advanced Context flag is set.
- **PML4[38:12]:** Pointer to base address of PML4 and selected when Legacy Context flag is set and 64b address support is selected.
- **PDP0[38:12]:** Pointer to one of the four page directory pointer (lowest) and defines the first 0-1GB of memory mapping.

**Note:** This is a guest physical address.
GFX PDP0/PML4/PASID Descriptor (Low Part)

<table>
<thead>
<tr>
<th>GFX_CTX_PDP0_L - GFX PDP0/PML4/PASID Descriptor (Low Part)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 04408h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>GFX PDP0/PML4/PASID Descriptor (Low Part)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

PDP0/PML4/PASID:
This register can contain three values which depend on the element descriptor definition.
PASID[19:0]: Populated in the first 20 bits of the register and selected when Advanced Context flag is set.
PML4[38:12]: Pointer to base address of PML4 and selected when Legacy Context flag is set and 64b address support is selected.
PDP0[38:12]: Pointer to one of the four page directory pointer (lowest) and defines the first 0-1GB of memory mapping.
Note: This is a guest physical address.
### GFX PDP1 Descriptor Register (High Part)

<table>
<thead>
<tr>
<th>GFX_CTX_PDP1_H - GFX PDP1 Descriptor Register (High Part)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x000000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 04414h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>GFX PDP1 Descriptor Register (High Part)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

Pointer to one of the four page directory pointer (lowest+1) and defines the first 1-2GB of memory mapping.

Note: This is a guest physical address.
### GFX PDP1 Descriptor Register (Low Part)

<table>
<thead>
<tr>
<th>GFX_CTX_PDP1_L - GFX PDP1 Descriptor Register (Low Part)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 04410h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>GFX PDP1 Descriptor Register (Low Part)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
</tbody>
</table>

Pointer to one of the four page directory pointer (lowest+1) and defines the first 1-2GB of memory mapping.

Note: This is a guest physical address.
## GFX PDP2 Descriptor Register (High Part)

<table>
<thead>
<tr>
<th>GFX_CTX_PDP2_H - GFX PDP2 Descriptor Register (High Part)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 0441Ch</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>GFX PDP2 Descriptor Register (High Part)</td>
</tr>
</tbody>
</table>

  * Default Value: 00000000h
  * Access: R/W

  Pointer to one of the four page directory pointer (lowest+2) and defines the first 2-3GB of memory mapping.

  Note: This is a guest physical address.
### GFX PDP2 Descriptor Register (Low Part)

<table>
<thead>
<tr>
<th><strong>GFX_CTX_PDP2_L</strong> - GFX PDP2 Descriptor Register (Low Part)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 04418h</td>
</tr>
</tbody>
</table>

**Description**

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>GFX PDP2 Descriptor Register (Low Part)</td>
</tr>
</tbody>
</table>

- **Default Value:** 00000000h
- **Access:** R/W

Pointer to one of the four page directory pointer (lowest+2) and defines the first 2-3GB of memory mapping.

Note: This is a guest physical address.
## GFX PDP3 Descriptor Register (High Part)

<table>
<thead>
<tr>
<th>GFX_CTX_PDP3_H - GFX PDP3 Descriptor Register (High Part)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 04424h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>GFX PDP3 Descriptor Register (High Part)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

Pointer to one of the four page directory pointer (lowest+3) and defines the first 3-4GB of memory mapping.

Note: This is a guest physical address.
## GFX PDP3 Descriptor Register (Low Part)

<table>
<thead>
<tr>
<th>GFX_CTX_PDP3_L - GFX PDP3 Descriptor Register (Low Part)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x000000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 04420h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>GFX PDP3 Descriptor Register (Low Part)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

Pointer to one of the four page directory pointer (lowest+3) and defines the first 3–4GB of memory mapping.

Note: This is a guest physical address.
**GGC - GGC**

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:15</td>
<td>RESERVED</td>
</tr>
<tr>
<td></td>
<td>Default Value:</td>
<td>00000h</td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td>Reserved</td>
<td></td>
</tr>
</tbody>
</table>
| 14    | VAMEN | Enables the use of the iGFX engines for Versatile Acceleration.  
1 - iGFX engines are in Versatile Acceleration Mode. Device 2 Class Code is 048000h.  
0 - iGFX engines are in iGFX Mode. Device 2 Class Code is 030000h.  |
|       | Default Value: | 0b |
|       | Access: | R/W Lock |
| 13:10 | RESERVED | |
|       | Default Value: | 0h |
|       | Access: | RO |
| 9:8   | GGMS | GTT Graphics Memory Size (GGMS):  
This field is used to select the amount of Main Memory that is pre-allocated to support the Internal Graphics Translation Table. The BIOS ensures that memory is pre-allocated only when Internal graphics is enabled.  
GSM is assumed to be a contiguous physical DRAM space with DSM, and BIOS needs to allocate a contiguous memory chunk. Hardware will derive the base of GSM from DSM only using the GSM size programmed in the register.  
Hardware functionality in case of programming this value to Reserved is not guaranteed.  
0h: No Preallocated Memory  
1h: 2MB of Preallocated Memory  
2h: 4MB of Preallocated Memory  
3h: 8MB of Preallocated Memory  |
<p>|       | Default Value: | 00b |
|       | Access: | R/W Lock |</p>
<table>
<thead>
<tr>
<th>7:3</th>
<th>GMS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Default Value: 00101b</td>
</tr>
<tr>
<td></td>
<td>Access: R/W Lock</td>
</tr>
</tbody>
</table>

**Graphics Mode Select (GMS):**
This field is used to select the amount of Main Memory that is pre-allocated to support the Internal Graphics device in VGA (non-linear) and Native (linear) modes. The BIOS ensures that memory is pre-allocated only when Internal graphics is enabled. Hardware does not clear or set any of these bits automatically based on IGD being disabled/enabled. BIOS Requirement: BIOS must not set this field to 0h if IVD (bit 1 of this register) is 0.

- **0h = 0MB**
- **1h = 32MB**
- **2h = 64MB**
- **3h = 96MB**
- **4h = 128MB**
- **5h = 160MB**
- **6h = 192MB**
- **7h = 224MB**
- **8h = 256MB**
- **9h = 288MB**
- **Ah = 320MB**
- **Bh = 352MB**
- **Ch = 384MB**
- **Dh = 416MB**
- **Eh = 448MB**
- **Fh = 480MB**
- **10h = 512MB**
- **11h = 8MB**
- **12h = 12MB**
- **13h = 16MB**
- **14h = 20MB**
- **15h = 24MB**
- **16h = 28MB**
- **17h = 36MB**
- **18h = 40MB**
- **19h = 44MB**
- **1Ah = 48MB**
- **1Bh = 52MB**
- **1Ch = 56MB**
- **1Dh = 60MB**
- **1Eh = Reserved**
- **1Fh = Reserved**

.....

- **20h: 1024MB (Not supported for CHV, BSW)**
## GGC - GGC

<table>
<thead>
<tr>
<th>Address</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>30h:1536MB (Not supported for CHV, BSW)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40h:2048MB (Not supported for CHV, BSW)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>80h:4096MB (Not supported for CHV, BSW)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>81h - FF:Reserved</td>
<td>Other = Reserved</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

When GMS != '0 (and VD=0):
Address[31:0] is compared with VGA memory range. (The VGA memory range is A_0000h to B_FFFFh.). If there is a match and MSE = 1 and MEMRD or MEMWR, the access will route as a Rmdwgamemen_cr cycle on the RMbus. If the RMbus returns a hit the GVD will select the command. As well, when 0 the GVD will check if scldown3_address[15:0] is one of the VGA IO register range. (The VGA IO range is 03B0h - 03BBh and 03C0h - 03DFh.) If there is a match and IOSE = 1 and the SCL command is either an IORD or IOWR, the GVD will initate a (VGA) register cycle on the RMbus. If the RMbus returns a hit the GVD will select the command.

When GMS == '0 :
No address compare will occur against VGA memory range or the VGA IO register range. Also, CC[15:8] is changed to 8'h80 from 8'h00

| 2 | RESERVED | Default Value: 0b | Access: RO |
| 1 | VGA_DISABLE | Default Value: 0b | Access: R/W Lock |
| 0 | GGCLCK | Default Value: 0b | Access: R/W Lock |

VGA Disable (VD):
0: Enable. Device 2 (IGD) claims VGA memory and IO cycles, the Sub-Class Code within Device 2 Class Code register is 00.
1: Disable. Device 2 (IGD) does not claim VGA cycles (Mem and IO), and the Sub-Class Code field within Device 2 function 0 Class Code register is 80.

BIOS Requirement: BIOS must not set this bit to 0 if the GMS field pre-allocates no memory. This bit MUST be set to 1 if Device 2 is disabled either via a fuse or fuse override (CAPID0[38] = 1) or via a register (DEVEN[3] = 0).

When set to 1b, this will lock all the bits in this register.
Global Clear Value Register 0

<table>
<thead>
<tr>
<th>GLOBAL_CLEAR_VALUE_0 - Global Clear Value Register 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: RenderCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Access: R/W</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 07700h</td>
</tr>
</tbody>
</table>

This register is to be used to program bits 31:0 of the Global clear value to be used for non MSRT render targets. This value will be used by pixel backend hardware in place of the 0/1 clear values, if the "Use Global Clear Value" bit is set.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Global Clear Value 0</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Contains bits 31:0 of the global clear value.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Clear value will be in native RT format.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Clear Value for 8/16/32 bpp RTs will be contained in this format. For the rest it will contain bits 31:0.</td>
</tr>
</tbody>
</table>
## Global Clear Value Register 1

<table>
<thead>
<tr>
<th>GLOBAL_CLEAR_VALUE_1 - Global Clear Value Register 1</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> RenderCS</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 07704h</td>
</tr>
</tbody>
</table>

This register is to be used to program bits 63:32 of the Global clear value to be used for non MSRT render targets. This value will be used by pixel backend hardware in place of the 0/1 clear values, if the "Use Global Clear Value" bit is set.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Global Clear Value 1</strong>&lt;br&gt;Contains bits 63:32 of the global clear value. Clear value will be in native RT format. This field will contain bits 63:32 of the clear value of 64 bpp and 128 bpp RTs.</td>
</tr>
</tbody>
</table>
Global Clear Value Register 2

<table>
<thead>
<tr>
<th>GLOBAL_CLEAR_VALUE_2 - Global Clear Value Register 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: RenderCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Access: R/W</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 07708h</td>
</tr>
</tbody>
</table>

This register is to be used to program bits 95:64 of the Global clear value to be used for non MSRT render targets. This value will be used by pixel backend hardware in place of the 0/1 clear values, if the “Use Global Clear Value” bit is set.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
</table>
| 0     | 31:0| **Global Clear Value 2**  
Contains bits 95:64 of the global clear value. Clear value will be in native RT format. Clear value will be in native RT format. This field will contain bits 95:64 of the clear value for 128 bpp RTs. |
Global Clear Value Register 3

| GLOBAL_CLEAR_VALUE_3 - Global Clear Value Register 3 |
|---------------------------------|---------------------------------|
| Register Space: MMIO: 0/2/0     | Project: CHV, BSW               |
| Source: RenderCS                | Default Value: 0x00000000       |
| Access: R/W                     | Size (in bits): 32              |
| Address: 0770Ch                  |                                 |

This register is to be used to program bits 127:96 of the Global clear value to be used for non MSRT render targets. This value will be used by pixel backend hardware in place of the 0/1 clear values, if the "Use Global Clear Value" bit is set.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Global Clear Value 3</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Contains bits 127:96 of the global clear value. Clear value will be in native RT format. This field will contain bits 127:96 of the clear value for 128 bpp RTs.</td>
</tr>
</tbody>
</table>
## Global Invalidation Register

<table>
<thead>
<tr>
<th>GLBLINVL - Global Invalidation Register</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 0B404h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:3</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved.</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td><strong>Cross sync read disable</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>lpconf_crs_sync_dis: Cross Sync Read Disable (CSRD).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cross Sync Read Disable (CSRD): Cross Sync Read Disable: upon a SYNC from HDC, follow with a write to cross SYNC Push and read to the same address. When set read is disabled.</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td><strong>Disables hashing function</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>lpconf_csr_l3bankidhashdis. (This bit needs to set corresponding bit lncf_csr_l3bankidhashdis in LNCF.)</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
</tbody>
</table>
# Global System Interrupt Routine

<table>
<thead>
<tr>
<th>EU_GLOBAL_SIP - Global System Interrupt Routine</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Access: R/W</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 0E42Ch</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:3</td>
<td><strong>Global SIP</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: GraphicsAddress[31:3]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Specifies the base address for System Interrupt Routine that over-rides the SIP set by the state (STATE_SIP).</td>
</tr>
<tr>
<td>2:1</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td><strong>Global SIP Enable</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The bit specifies if the System Routine starts from the Global SIP provided by the DW OR the SIP provided by the state (STATE_SIP)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>SIP used is from STATE_EIP</td>
</tr>
<tr>
<td>1</td>
<td>SIP used is from MMIO register</td>
</tr>
</tbody>
</table>
**GMADR_LSB**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>PCI: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x0000000C</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>00018h</td>
</tr>
</tbody>
</table>

Gfx Aperture location. 
GMADR is a Prefetchable range in order to apply USWC attribute (from the processor point of view) to that range. The USWC attribute is used by the processor for write combining. Accesses to this range will be translated to DRAM Physical memory addresses. Fence registers may be used to sub-divide this range and allow tiled surfaces (determined by fence registers). The supported sizes are determined by the MSAC register.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31</td>
<td><strong>ADMSK4096</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td><strong>ADMSK2048</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td>29</td>
<td><strong>ADMSK1024</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td><strong>ADMSK512</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
</tbody>
</table>

4096MB Address Mask (ADMSK4096): Locked with MSAC.APSZ[4]. See MSAC (Dev2, Func 0, offset 62h) for details.

2048MB Address Mask (ADMSK2048): Locked with MSAC.APSZ[3]. See MSAC (Dev2, Func 0, offset 62h) for details.

1024MB Address Mask (ADMSK1024): Locked with MSAC.APSZ[2]. See MSAC (Dev2, Func 0, offset 62h) for details.

512MB Address Mask (ADMSK512): Locked with MSAC.APSZ[1]. See MSAC (Dev2, Func 0, offset 62h) for details.
<table>
<thead>
<tr>
<th>Offset</th>
<th>Field</th>
<th>Default Value</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>27</td>
<td>ADMSK256</td>
<td>0b</td>
<td>R/W</td>
<td>256MB Address Mask (ADMSK256): Locked with MSAC.APSZ[0]. See MSAC (Dev2, Func 0, offset 62h) for details.</td>
</tr>
<tr>
<td>26:4</td>
<td>Reserved</td>
<td>000000h</td>
<td>RO</td>
<td>Address Mask (ADM): Hardwired to 0s to indicate at least 128MB address range.</td>
</tr>
<tr>
<td>3</td>
<td>PREFMEM</td>
<td>1b</td>
<td>RO</td>
<td>Prefetchable Memory (PREFMEM): Hardwired to 1 to enable prefetching.</td>
</tr>
<tr>
<td>2:1</td>
<td>MEMTYP</td>
<td>10b</td>
<td>RO</td>
<td>Memory Type (MEMTYP): 00: To indicate 32 bit base address 01: Reserved 10: To indicate 64 bit base address 11: Reserved</td>
</tr>
<tr>
<td>0</td>
<td>Reserved</td>
<td>0b</td>
<td>RO</td>
<td>Memory/IO Space (MIOS): Hardwired to 0 to indicate memory space.</td>
</tr>
</tbody>
</table>
**GMADR_MSB**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>PCI: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0001Ch</td>
</tr>
</tbody>
</table>

Gfx Aperture location.
GMADR is a Prefetchable range in order to apply USWC attribute (from the processor point of view) to that range. The USWC attribute is used by the processor for write combining.
Accesses to this range will be translated to DRAM Physical memory addresses. Fence registers may be used to sub-divide this range and allow tiled surfaces (determined by fence registers).
The supported sizes are determined by the MSAC register.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:4</td>
<td>MBA_MSB28</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Set by the OS, these bits correspond to address signals [63:39]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Made them spare RW bits.</td>
</tr>
<tr>
<td>3:0</td>
<td>MBA</td>
<td>Memory Base Address (MBA)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Set by the OS, these bits correspond to address signals [35:32]</td>
</tr>
</tbody>
</table>
## GMBC Message Register

### GMBCMSG - GMBC Message Register

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>08500h</td>
</tr>
</tbody>
</table>

### GMBC Message Register

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:12</td>
<td><strong>RSVD</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td><strong>Context Restore Message</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Context Restore Message</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 = Context Restore Message From CS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 = Not a Message, GMBC clears when sending ACK to CS.</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td><strong>MBC BUSY ACK Message</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 = Busy Ack message from GPMG</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 = IDLE (non-busy) Ack message from GPMG</td>
</tr>
<tr>
<td>9:7</td>
<td></td>
<td><strong>RSVD</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td><strong>Block Message</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Block Message from GPMG</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 = Block Outbound GAM Traffic and wait for outstanding GAM write requests to be GO’d and reads to return</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 = Unblock Outbound GAM traffic</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td><strong>RSVD</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td><strong>Arbitration request/release ACK</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong></td>
</tr>
</tbody>
</table>
## GMBCMSG - GMBC Message Register

<table>
<thead>
<tr>
<th>Bit</th>
<th>Field</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td><strong>RSVD</strong></td>
<td><strong>RO</strong></td>
<td>Reserved</td>
</tr>
<tr>
<td>2</td>
<td><strong>Fuse Fetch Message</strong></td>
<td><strong>R/W</strong></td>
<td>Fuse Fetch message: written to 1 by GPMG to indicate GMBC should perform a fuse fetch. Cleared by GMBC when done with the fuse fetch.</td>
</tr>
<tr>
<td>1:0</td>
<td><strong>RSVD</strong></td>
<td><strong>RO</strong></td>
<td>Reserved</td>
</tr>
</tbody>
</table>

- **Global Arb request ACK message:**
  - 1 = GPMG ACK of GMBC Global Arb Update Request.
  - 0 = GPMG ACK of GMBC Global Arb Release Request.
### GO Messaging Register for GAMunit

**MSG_GO_GAM - GO Messaging Register for GAMunit**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>16</td>
</tr>
<tr>
<td>Address:</td>
<td>08028h</td>
</tr>
</tbody>
</table>

Message registers have bit-wise masking applied for writes. The register consists of 16 bits of data in [15:0], and 16 bits of corresponding masks in [31:16]. In order to set bit0, for example, the data would be 0x0001_0001. In order to clear bit0, for example, the data would be 0x0000_0001. Note that mask bit is the data bit offset + 16.

Message registers are protected from non-GT writes via the Message Channel.

**GA* Response to Allow Graphics Cycles to Read/Write from Memory**
1'b0 : No gfx cycles allowed to memory <default>
1'b1 : Allow gfx cycles to memory
gpm currently only ever sends the same GO request for all agents; either all is go=0, or all is go=1.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Access:</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>15:7</td>
<td><strong>Reserved</strong></td>
<td>RO</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td><em><em>GA</em> Response to Allow Wi-Di Graphics Cycles to Read/Write from Memory</em>*</td>
<td>R/W</td>
<td>[6] Controls Wi-Di Cycles (winunit)</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td><strong>Reserved</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td><em><em>GA</em> Response to Allow VEBox Graphics Cycles to Read/Write from Memory</em>*</td>
<td>R/W</td>
<td>[3] Controls VEBox Cycles (vecsunit)</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td><em><em>GA</em> Response to Allow Media1 Graphics Cycles to Read/Write from Memory</em>*</td>
<td>R/W</td>
<td>[2] Controls Media1 Cycles (vcs1unit)</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td><em><em>GA</em> Response to Allow Media0 Graphics Cycles to Read/Write from Memory</em>*</td>
<td>R/W</td>
<td>[1] Controls Media0 Cycles (vcs0unit)</td>
</tr>
</tbody>
</table>
### MSG_GO_GAM - GO Messaging Register for GAMunit

<table>
<thead>
<tr>
<th></th>
<th>GA* Response to Allow Render Graphics Cycles to Read/Write from Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td>[0] Controls Render Cycles (csunit)</td>
</tr>
</tbody>
</table>
### Go Protocol GAM Request

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td><strong>Mask Bits</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved.</td>
</tr>
<tr>
<td>15</td>
<td>GO_PROTOCOL_GAM_REQUEST15</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Go Protocol Request Reasons: 1'b0: Engine will NOT be resetting. 1'b1: Engine will be resetting (FLR/RC6 Enter or Soft Reset). Preparation for FLR (device) reset (cdevrst_b).</td>
</tr>
<tr>
<td>14</td>
<td>GO_PROTOCOL_GAM_REQUEST14</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Go Protocol Request Reasons: 1'b0: Engine will NOT be resetting. 1'b1: Engine will be resetting (FLR/RC6 Enter or Soft Reset). Preparation for Media1 reset (vcs1unit).</td>
</tr>
<tr>
<td>13</td>
<td>GO_PROTOCOL_GAM_REQUEST13</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Go Protocol Request Reasons: 1'b0: Engine will NOT be resetting. 1'b1: Engine will be resetting (FLR/RC6 Enter or Soft Reset). Preparation for Wi-Di reset (winunit).</td>
</tr>
</tbody>
</table>
### GO_GAM_REQ - Go Protocol GAM Request

<table>
<thead>
<tr>
<th>12</th>
<th>GO_PROTOCOL_GAM_REQUEST12</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Value:</td>
<td>0b</td>
</tr>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Go Protocol Request Reasons:</td>
<td></td>
</tr>
<tr>
<td>1'b0: Engine will NOT be resetting.</td>
<td></td>
</tr>
<tr>
<td>1'b1: Engine will be resetting (FLR/RC6 Enter or Soft Reset).</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>11</th>
<th>GO_PROTOCOL_GAM_REQUEST11</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Value:</td>
<td>0b</td>
</tr>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Go Protocol Request Reasons:</td>
<td></td>
</tr>
<tr>
<td>1'b0: Engine will NOT be resetting.</td>
<td></td>
</tr>
<tr>
<td>1'b1: Engine will be resetting (FLR/RC6 Enter or Soft Reset).</td>
<td></td>
</tr>
<tr>
<td>Preparation for Blitter reset (bcunit).</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>10</th>
<th>GO_PROTOCOL_GAM_REQUEST10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Value:</td>
<td>0b</td>
</tr>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Go Protocol Request Reasons:</td>
<td></td>
</tr>
<tr>
<td>1'b0: Engine will NOT be resetting.</td>
<td></td>
</tr>
<tr>
<td>1'b1: Engine will be resetting (FLR/RC6 Enter or Soft Reset).</td>
<td></td>
</tr>
<tr>
<td>Preparation for VEBox reset (vecsunit).</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>9</th>
<th>GO_PROTOCOL_GAM_REQUEST9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Value:</td>
<td>0b</td>
</tr>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Go Protocol Request Reasons:</td>
<td></td>
</tr>
<tr>
<td>1'b0: Engine will NOT be resetting.</td>
<td></td>
</tr>
<tr>
<td>1'b1: Engine will be resetting (FLR/RC6 Enter or Soft Reset).</td>
<td></td>
</tr>
<tr>
<td>Preparation for Media0 reset (vcs0unit).</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>8</th>
<th>GO_PROTOCOL_GAM_REQUEST8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Value:</td>
<td>0b</td>
</tr>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Go Protocol Request Reasons:</td>
<td></td>
</tr>
<tr>
<td>1'b0: Engine will NOT be resetting.</td>
<td></td>
</tr>
<tr>
<td>1'b1: Engine will be resetting (FLR/RC6 Enter or Soft Reset).</td>
<td></td>
</tr>
<tr>
<td>Preparation for Render reset (csunit).</td>
<td></td>
</tr>
</tbody>
</table>
### GO_GAM_REQ - Go Protocol GAM Request

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>Reserved</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>GO_PROTOCOL_GAM_REQUEST6</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>GPM to GAM Go Protocol Request.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0: No graphic cycles allowed to memory (default).</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1: Allow graphic cycles to memory.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Controls Wi-Di Cycles (winunit).</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>GPM currently only ever sends the same GO request for all agents; either all is go=0, or all is go=1.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Reserved</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>GO_PROTOCOL_GAM_REQUEST4</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>GPM to GAM Go Protocol Request.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0: No graphic cycles allowed to memory (default).</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1: Allow graphic cycles to memory.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Controls Blitter Cycles (bcsunit).</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>GPM currently only ever sends the same GO request for all agents; either all is go=0, or all is go=1.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>GO_PROTOCOL_GAM_REQUEST3</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>GPM to GAM Go Protocol Request.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0: No graphic cycles allowed to memory (default).</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1: Allow graphic cycles to memory.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Controls VEBox Cycles (vecsunit).</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>GPM currently only ever sends the same GO request for all agents; either all is go=0, or all is go=1.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>GO_PROTOCOL_GAM_REQUEST2</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>GPM to GAM Go Protocol Request.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0: No graphic cycles allowed to memory (default).</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1: Allow graphic cycles to memory.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Controls Media1 Cycles (vcs1unit).</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>GPM currently only ever sends the same GO request for all agents; either all is go=0, or all is go=1.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>GO_PROTOCOL_GAM_REQUEST1</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>GPM to GAM Go Protocol Request.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0: No graphic cycles allowed to memory (default).</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1: Allow graphic cycles to memory.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## GO_GAM_REQ - Go Protocol GAM Request

<table>
<thead>
<tr>
<th>0</th>
<th>GO_PROTOCOL_GAM_REQUEST0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Value:</td>
<td>0b</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
</tbody>
</table>

GPM to GAM Go Protocol Request.
0: No graphic cycles allowed to memory (default).
1: Allow graphic cycles to memory.
Controls Media0 Cycles (vcs0unit).
GPM currently only ever sends the same GO request for all agents; either all is go=0, or all is go=1.
## GPA to HPA Translation Request

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td><strong>Mask Bits</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Mask Bits act as Write Enables for the bits[15:0] of this register.</td>
</tr>
<tr>
<td>15</td>
<td>GPA to HPA Translation Request 15</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default Value: 0b</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td>For Future Use.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>This bit is self clear.</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>GPA to HPA Translation Request 14</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default Value: 0b</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td>For Future Use.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>This bit is self clear.</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>GPA to HPA Translation Request 13</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default Value: 0b</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td>For Future Use.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>This bit is self clear.</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>GPA to HPA Translation Request 12</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default Value: 0b</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td>For Future Use.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>This bit is self clear.</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>GPA to HPA Translation Request 11</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default Value: 0b</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td>For Future Use.</td>
<td></td>
</tr>
<tr>
<td>GPA2HPAR - GPA to HPA Translation Request</td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>This bit is self clear.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>10 GPA to HPA Translation Request 10</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default Value:</td>
<td>0b</td>
<td></td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>For Future Use.</td>
<td>This bit is self clear.</td>
<td></td>
</tr>
<tr>
<td><strong>9 GPA to HPA Translation Request 9</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default Value:</td>
<td>0b</td>
<td></td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>For Future Use.</td>
<td>This bit is self clear.</td>
<td></td>
</tr>
<tr>
<td><strong>8 GPA to HPA Translation Request 8</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default Value:</td>
<td>0b</td>
<td></td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>For Future Use.</td>
<td>This bit is self clear.</td>
<td></td>
</tr>
<tr>
<td><strong>7 GPA to HPA Translation Request 7</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default Value:</td>
<td>0b</td>
<td></td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>For Future Use.</td>
<td>This bit is self clear.</td>
<td></td>
</tr>
<tr>
<td><strong>6 GPA to HPA Translation Request 6</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default Value:</td>
<td>0b</td>
<td></td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>For Future Use.</td>
<td>This bit is self clear.</td>
<td></td>
</tr>
<tr>
<td><strong>5 GPA to HPA Translation Request 5</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default Value:</td>
<td>0b</td>
<td></td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>For Future Use.</td>
<td>This bit is self clear.</td>
<td></td>
</tr>
<tr>
<td><strong>4 GPA to HPA Translation Request 4</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default Value:</td>
<td>0b</td>
<td></td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>For Future Use.</td>
<td>This bit is self clear.</td>
<td></td>
</tr>
<tr>
<td><strong>3 GPA to HPA Translation Request 3</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default Value:</td>
<td>0b</td>
<td></td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
<td></td>
</tr>
</tbody>
</table>
### GPA2HPAR - GPA to HPA Translation Request

<table>
<thead>
<tr>
<th>GPA to HPA Translation Request</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td>1</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td>0</td>
<td>0b</td>
<td>R/W</td>
</tr>
</tbody>
</table>

**For Future Use.**

This bit is self clear.

**2 GPA to HPA Translation Request 2**

Default Value: 0b

Access: R/W

Bit[2]: A request for GPA to HPA translation. Note that GPA register should have been written prior to sending the message for the translation. Mask bit[18] needs to be enabled to program the register.

This bit is self clear.

**1 GPA to HPA Translation Request 1**

Default Value: 0b

Access: R/W

For Future Use.

This bit is self clear.

**0 GPA to HPA Translation Request 0**

Default Value: 0b

Access: R/W

For Future Use.

This bit is self clear.
### GPA2HPAV - GPA value for GPA to HPA Translation

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>04210h</td>
</tr>
<tr>
<td>DWord</td>
<td>Bit</td>
</tr>
<tr>
<td>0</td>
<td>31:0</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## GPGPU Context Restore Request To TDL

<table>
<thead>
<tr>
<th><strong>GPGPU_CTX_RESTORE - GPGPU Context Restore Request To TDL</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Access:</strong> WO</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 0E4CCh</td>
</tr>
<tr>
<td><strong>Name:</strong> GPGPU Context Restore Request To TDL Slice 0 SubSlice 0</td>
</tr>
<tr>
<td><strong>ShortName:</strong> GPGPU_CTX_RESTORE_S0_SS0</td>
</tr>
<tr>
<td><strong>Valid Projects:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Address:</strong> 0E5CCh</td>
</tr>
<tr>
<td><strong>Name:</strong> GPGPU Context Restore Request To TDL Slice 0 SubSlice 1</td>
</tr>
<tr>
<td><strong>ShortName:</strong> GPGPU_CTX_RESTORE_S0_SS1</td>
</tr>
<tr>
<td><strong>Valid Projects:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Address:</strong> 0E6CCh</td>
</tr>
<tr>
<td><strong>Name:</strong> GPGPU Context Restore Request To TDL Slice 0 SubSlice 2</td>
</tr>
<tr>
<td><strong>ShortName:</strong> GPGPU_CTX_RESTORE_S0_SS2</td>
</tr>
<tr>
<td><strong>Valid Projects:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Address:</strong> 0E4DCh</td>
</tr>
<tr>
<td><strong>Name:</strong> GPGPU Context Restore Request To TDL Slice 1 SubSlice 0</td>
</tr>
<tr>
<td><strong>ShortName:</strong> GPGPU_CTX_RESTORE_S1_SS0</td>
</tr>
<tr>
<td><strong>Valid Projects:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Address:</strong> 0E5DCh</td>
</tr>
<tr>
<td><strong>Name:</strong> GPGPU Context Restore Request To TDL Slice 1 SubSlice 1</td>
</tr>
<tr>
<td><strong>ShortName:</strong> GPGPU_CTX_RESTORE_S1_SS1</td>
</tr>
<tr>
<td><strong>Valid Projects:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Address:</strong> 0E6DCh</td>
</tr>
<tr>
<td><strong>Name:</strong> GPGPU Context Restore Request To TDL Slice 1 SubSlice 2</td>
</tr>
<tr>
<td><strong>ShortName:</strong> GPGPU_CTX_RESTORE_S1_SS2</td>
</tr>
<tr>
<td><strong>Valid Projects:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Address:</strong> 0E4ECh</td>
</tr>
<tr>
<td><strong>Name:</strong> GPGPU Context Restore Request To TDL Slice 2 SubSlice 0</td>
</tr>
<tr>
<td><strong>ShortName:</strong> GPGPU_CTX_RESTORE_S2_SS0</td>
</tr>
<tr>
<td><strong>Valid Projects:</strong> CHV, BSW</td>
</tr>
</tbody>
</table>
## GPGPU_CTX_RESTORE - GPGPU Context Restore Request To TDL

<table>
<thead>
<tr>
<th>Address</th>
<th>0E5ECh</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>GPGPU Context Restore Request To TDL Slice 2 SubSlice 1</td>
</tr>
<tr>
<td>ShortName</td>
<td>GPGPU_CTX_RESTORE_S2_SS1</td>
</tr>
<tr>
<td>Valid Projects</td>
<td>CHV, BSW</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Address</th>
<th>0E6ECh</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>GPGPU Context Restore Request To TDL Slice 2 SubSlice 2</td>
</tr>
<tr>
<td>ShortName</td>
<td>GPGPU_CTX_RESTORE_S2_SS2</td>
</tr>
<tr>
<td>Valid Projects</td>
<td>CHV, BSW</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

Format: MBZ
## GPGPU Context Save Request To TDL

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>WO</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0E4D8h</td>
</tr>
<tr>
<td>Valid Projects:</td>
<td>CHV, BSW</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
</tbody>
</table>

**DWord**  
**Bit**  
**Description**  

**Reserved**  
**Format:** MBZ
## GPGPU Dispatch Dimension X

<table>
<thead>
<tr>
<th>GPGPU DISPATCHDIMX - GPGPU Dispatch Dimension X</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: RenderCS</td>
</tr>
<tr>
<td>Default Value: 0x0000000000</td>
</tr>
<tr>
<td>Access: R/W</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 02500h</td>
</tr>
<tr>
<td>DWord</td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>0, FFFFFFFFh</td>
<td></td>
<td>CHV, BSW</td>
</tr>
</tbody>
</table>
### GPGPU Dispatch Dimension Y

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Dispatch Dimension Y</strong></td>
</tr>
</tbody>
</table>

Format: U32

The number of thread groups to be dispatched in the Y dimension (max y + 1)

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>0, FFFFFFFFh</td>
<td></td>
<td>CHV, BSW</td>
</tr>
</tbody>
</table>
# GPGPU Dispatch Dimension Z

<table>
<thead>
<tr>
<th><strong>GPGPU DISPATCHDIMZ - GPGPU Dispatch Dimension Z</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: RenderCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Access: R/W</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 02508h</td>
</tr>
</tbody>
</table>

**Description**

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

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>0, FFFFFFFFh</td>
<td></td>
<td>CHV, BSW</td>
</tr>
</tbody>
</table>
# GPU_Ticks_Counter

<table>
<thead>
<tr>
<th><strong>GPU_TICKS - GPU_Ticks_Counter</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 02910h</td>
</tr>
<tr>
<td><strong>Valid Projects:</strong> CHV, BSW</td>
</tr>
</tbody>
</table>

Reading this register returns the live value of the GPU ticks counter that is sampled and included in the performance counter report header.

<table>
<thead>
<tr>
<th><strong>DWORD</strong></th>
<th><strong>Bit</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Considerations</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format:</strong> U32</td>
</tr>
</tbody>
</table>

This 32-bit field returns bits 31:0 of the GPU tick counter value when read. Note that there is no "latch and hold" mechanism for performance counters when they are accessed through MMIO, so the value returned from this register may be different on back-to-back reads.
### Graphics Device Reset Control

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:7</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Reserved:</td>
<td>MBZ</td>
</tr>
<tr>
<td>5</td>
<td><strong>Reserved</strong></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td><strong>Initiate Graphics Vebox Soft Reset</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>R/W Set</td>
</tr>
<tr>
<td></td>
<td>Graphics VEbox Soft-Reset Control:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>’1’ : Initiate a graphics Vebox domain reset.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Cleared by CP once the reset is complete</td>
<td></td>
</tr>
<tr>
<td></td>
<td>’0’ : N/A</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Once set, clearing of this bit has no effect on CP. Only CP can reset this bit.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Note: This is a non-posted register.</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td><strong>Initiate Graphics Blitter Soft Reset</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>R/W Set</td>
</tr>
<tr>
<td></td>
<td>Graphics Blitter Soft-Reset Control:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>’1’ : Initiate a graphics blitter domain reset.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Cleared by CP once the reset is complete</td>
<td></td>
</tr>
<tr>
<td></td>
<td>’0’ : N/A</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Once set, clearing of this bit has no effect on CP. Only CP can reset this bit.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Note: This is a non-posted register.</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td><strong>Initiate Graphics Media Soft Reset</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>R/W Set</td>
</tr>
<tr>
<td></td>
<td>Graphics Media Soft-Reset Control:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>’1’ : Initiate a graphics media 0 domain reset.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Cleared by CP once the reset is complete</td>
<td></td>
</tr>
<tr>
<td></td>
<td>’0’ : N/A</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Once set, clearing of this bit has no effect on CP. Only CP can reset this bit.</td>
<td></td>
</tr>
<tr>
<td>GDRST - Graphics Device Reset Control</td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Note:</strong> This is a non-posted register.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

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

**GFX_MODE - Graphics Mode Register**

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space:</td>
<td>MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>RenderCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000 [CHV:B, CHV:C, CHV:k]</td>
</tr>
<tr>
<td></td>
<td>0x00000800 [CHV:A]</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Trusted Type:</td>
<td>1</td>
</tr>
<tr>
<td>Address:</td>
<td>0229Ch</td>
</tr>
<tr>
<td>Valid Projects:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Address:</td>
<td>1229Ch-1229Fh</td>
</tr>
<tr>
<td>Name:</td>
<td>Graphics Mode Register</td>
</tr>
<tr>
<td>ShortName:</td>
<td>GFX_MODE_VCSUNIT0</td>
</tr>
<tr>
<td>Address:</td>
<td>1A29Ch-1A29Fh</td>
</tr>
<tr>
<td>Name:</td>
<td>Graphics Mode Register</td>
</tr>
<tr>
<td>ShortName:</td>
<td>GFX_MODE_VECSUNIT</td>
</tr>
<tr>
<td>Address:</td>
<td>1C29Ch-1C29Fh</td>
</tr>
<tr>
<td>Name:</td>
<td>Graphics Mode Register</td>
</tr>
<tr>
<td>ShortName:</td>
<td>GFX_MODE_VCSUNIT1</td>
</tr>
<tr>
<td>Address:</td>
<td>2229Ch-2229Fh</td>
</tr>
<tr>
<td>Name:</td>
<td>Graphics Mode Register</td>
</tr>
<tr>
<td>ShortName:</td>
<td>GFX_MODE_BCSUNIT</td>
</tr>
</tbody>
</table>

**Description**

This register contains a control bit for the new execlist and 2-level PPGTT functions.

Default Value = 00002800h

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td><strong>Mask Bits</strong>&lt;br&gt;Format: Mask[15:0]&lt;br&gt;Must be set to modify corresponding bit in Bits 15:0. (All implemented bits)</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td><strong>Execlist Enable</strong>&lt;br&gt;Mask: MMIO#31&lt;br&gt;When set, software can utilize the execlist registers to load a context into hardware. MI_SET_CONTEXT and MI_ARB_CHECK commands will be converted to NOOP if parsed. When this bit is clear, the Execlist mechanism cannot be used. The context must be loaded via MI_SET_CONTEXT and the ring must be loaded via MMIO access.</td>
</tr>
</tbody>
</table>
## GFX_MODE - Graphics Mode Register

### Programming Notes
This bit is *not* intended to be changed dynamically. Changing the value of this bit while rendering is in progress will have UNDEFINED results. This bit should be changed only *after a full reset* and *before submitting any* commands to the device.

<table>
<thead>
<tr>
<th>Bit</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>Reserved</td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>13</td>
<td>Flush TLB invalidation Mode</td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: U1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This field controls the invalidation if the TLB cache inside the hardware. When enabled this bit limits the invalidation of the TLB only to batch buffer boundaries, to pipe_control commands which have the TLB invalidation bit set and sync flushes. If disabled, the TLB caches are flushed for every full flush of the pipeline.</td>
</tr>
<tr>
<td>12</td>
<td>Reserved</td>
<td>Project: All</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
<tr>
<td>11</td>
<td>Reserved</td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
<tr>
<td>11</td>
<td>Replay Mode</td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: U1 Context Switch Granularity</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This field controls the granularity of the replay mechanism when coming back into a previously preempted context.</td>
</tr>
<tr>
<td></td>
<td>Value</td>
<td>Name</td>
</tr>
<tr>
<td>1h</td>
<td></td>
<td>Object Level Preemption</td>
</tr>
<tr>
<td>0h</td>
<td></td>
<td>mid-cmdbuffer preemption</td>
</tr>
</tbody>
</table>

### Programming Notes
A fixed function pipe flush is required before modifying this field.

The replay mode must be set to 0.

<table>
<thead>
<tr>
<th>Bit</th>
<th>Reserved</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>Project: All</td>
</tr>
</tbody>
</table>
### GFX_MODE - Graphics Mode Register

#### Per-Process GTT Enable

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>PPGTT Disable</td>
<td>When clear, the Global GTT will be used to translate memory access from designated commands and for commands that select the PPGTT as their translation space.</td>
</tr>
<tr>
<td>1h</td>
<td>PPGTT Enable</td>
<td>When set, the PPGTT will be used to translate memory access from designated commands and for commands that select the PPGTT as their translation space.</td>
</tr>
</tbody>
</table>

**Programming Notes**

This bit is used for enabling PPGTT access in Ring Buffer mode of scheduling. Privilege field in context descriptor states the same in Execlist Mode of scheduling. This field should be set before programming PDP0/1/2/3 registers in order to set the PPGTT translation of memory access.

Programming this bit doesn’t enable or disable the PPGTT translation of memory access immediately; the change comes in to affect only when the Page Directory registers are programmed. Programming this bit must be followed by programming Page Directory Registers in order to enable or disable the PPGTT translation of memory access.

#### 64Bit Virtual Addressing Enable

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>64Bit Virtual Address Disable [Default]</td>
<td>When clear indicates GFX operating in 32bit Virtual Addressing for PPGTT based memory access.</td>
</tr>
</tbody>
</table>

**Programming Notes**

This bit is only valid when PPGTT is enabled in ring buffer mode of scheduling. Context Descriptor has a similar bit to control 64bit virtual addressing in execlist mode of scheduling. Whether this field is set or clear virtual addresses translated through GGTT are always 32Bit. This field should be programmed before enabling PPGTT access. When this field is not set or for GGTT virtual addresses, Graphics Address [47:32] field of any commands or register exercised by SW should be programmed to 0x0.

64 Bit Virtual Addressing is not supported on CHV, BSW and must not be set.
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Project</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>6:5</td>
<td>Reserved</td>
<td>CHV, BSW</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Reserved</td>
<td>CHV, BSW</td>
<td></td>
</tr>
<tr>
<td>3:1</td>
<td>Reserved</td>
<td>CHV, BSW</td>
<td>MBZ</td>
</tr>
<tr>
<td>0</td>
<td>Privilege Check Disable</td>
<td>CHV, BSW</td>
<td>Enable</td>
</tr>
</tbody>
</table>

This field when set, disables Privilege Violation checks on non-privileged batch buffers. When set Privileged commands are allowed to be executed from non-privileged batch buffers.

**Workaround**

Workaround: Irrespective of "Privilege Check Disable" bit set, HW enforces chained or second level batch buffer "Address Space Indicator" to be PPGTT if the parent batch buffer Address Space Indicator is PPGTT.
### GS Domain Clock Gate Control Register

**GSCKGCTL - GS Domain Clock Gate Control Register**

<table>
<thead>
<tr>
<th>Register Space</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MMIO: 0/2/0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CHV, BSW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PRM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x00000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Address:** 09028h

Gunit Graphics Slow Clock Domain Clock Gating Control Register

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:29</td>
<td><strong>RSVD</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: <strong>RO</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>RSVD</strong></td>
</tr>
<tr>
<td>28:24</td>
<td><strong>GMBC Clock Gate IDLE Count</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: <strong>R/W</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>GMBC IDLE Timer Count Value - bits[27:24]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 = GMBC Idle timer disabled</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1-15 = After GMBC indicates idle, wait this many clocks before gating.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit [28] is additional Reserved R/W</td>
</tr>
<tr>
<td>23:20</td>
<td><strong>GPMG Clock Gate IDLE Counter</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: <strong>R/W</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>GPMG idle timer. GPMG uses this as a its idle counter pre-load value for clock gating</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 = Idle timer disabled</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 - 15 = After unit indicates its idle, wait this many clocks before gating.</td>
</tr>
<tr>
<td>19:16</td>
<td><strong>GSIDLECNT</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: <strong>R/W</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic GUNIT GSCLK domain idle timer. GUNIT GS fubs (other than GMBC) use this as a counter pre-load value for clock gating</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 = Idle timer disabled</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 - 15 = After a unit indicates idle, wait this many clocks before gating.</td>
</tr>
<tr>
<td>15:13</td>
<td><strong>RSVD</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: <strong>RO</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td>12</td>
<td><strong>GDTCKGEN</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: <strong>R/W</strong></td>
</tr>
</tbody>
</table>
**GSCKGCTL - GS Domain Clock Gate Control Register**

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
<th>Access</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>GDT Clock Gating Enable</td>
<td>0 = Disable clock gating. 1 = Enable clock gating.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GMBCPCKDIS</td>
<td>GMBC Performance Monitor Clock Gating Disable</td>
<td>R/W</td>
<td>0 = Enable clock gating 1 = Disable clock gating</td>
</tr>
<tr>
<td>GMBCCKGD</td>
<td>GMBC Clock Gating Disable: controls non-performance monitoring related clocks in the GMBC</td>
<td>R/W</td>
<td>0 = Enable clock gating 1 = Disable clock gating</td>
</tr>
<tr>
<td>GPMGCKD</td>
<td>GPMG Clock Gating Disable</td>
<td>R/W</td>
<td>0 = Enable clock gating 1 = Disable clock gating</td>
</tr>
<tr>
<td>G3DRDCKGD</td>
<td>G3D Read Return Clock Gating Disable</td>
<td>R/W</td>
<td>0 = Enable clock gating 1 = Disable clock gating</td>
</tr>
<tr>
<td>G3DOUTCKGD</td>
<td>G3D Outbound Clock Gating Disable: controls clocks associated with outbound requests.</td>
<td>R/W</td>
<td>0 = Enable clock gating 1 = Disable clock gating</td>
</tr>
<tr>
<td>GMCFGCKGD</td>
<td>GMCFG Clock Gating Disable</td>
<td>R/W</td>
<td>0 = Enable clock gating 1 = Disable clock gating</td>
</tr>
<tr>
<td>GKEYSCKGD</td>
<td></td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>GSCKGCTL - GS Domain Clock Gate Control Register</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---------------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td><strong>GCCBGSGDIS</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>GCCBGS Clock Gating Disable</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 = Enable clock gating</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 = Disable clock gating</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3:0</td>
<td><strong>RSVD</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access: RO</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Reserved</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
GS Invocation Counter

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>RenderCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000, 0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>64</td>
</tr>
<tr>
<td>Trusted Type:</td>
<td>1</td>
</tr>
</tbody>
</table>

Address: 02328h

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>63:32</td>
<td><strong>GS Invocation Count UDW</strong>&lt;br&gt;Number of objects that are dispatched as a geometry shader threads invoked by the GS stage. Updated only when <strong>Statistics Enable</strong> is set in GS_STATE (see the Geometry Shader Chapter in the 3D Volume.)</td>
</tr>
<tr>
<td>31:0</td>
<td></td>
<td><strong>GS Invocation Count LDW</strong>&lt;br&gt;Number of objects that are dispatched as a geometry shader threads invoked by the GS stage. Updated only when <strong>Statistics Enable</strong> is set in GS_STATE (see the Geometry Shader Chapter in the 3D Volume.)</td>
</tr>
</tbody>
</table>
### GS Primitives Counter

<table>
<thead>
<tr>
<th><strong>GS_PRIMITIVES_COUNT</strong> - GS Primitives Counter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: RenderCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000, 0x00000000</td>
</tr>
<tr>
<td>Access: R/W</td>
</tr>
<tr>
<td>Size (in bits): 64</td>
</tr>
<tr>
<td>Trusted Type: 1</td>
</tr>
<tr>
<td>Address: 02330h</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>63:32</td>
<td><strong>GS Primitives Count UDW</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Total number of primitives output by the geometry stage. Updated only when</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Statistics Enable is set in GS_STATE (see the Geometry Shader Chapter in the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3D Volume.</td>
</tr>
<tr>
<td></td>
<td>31:0</td>
<td><strong>GS Primitives Count LDW</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Total number of primitives output by the geometry stage. Updated only when</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Statistics Enable is set in GS_STATE (see the Geometry Shader Chapter in the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3D Volume.</td>
</tr>
</tbody>
</table>
## GT_CR_POWER_METER_CTRL

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>31</td>
<td><strong>Power Meter and Push Bus Control Lock</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Controls whether power meter weights and other push bus control registers are writeable.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 : Power Meter weights and control registers are writeable.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1: Writes to Power Meter weights and control registers are blocked.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Lock bit cannot be cleared without cold reset.</td>
</tr>
<tr>
<td>30:7</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td><strong>GSclk Domain GTI Baseline Energy Count Enable</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Controls whether the energy counter for the GTI power well, counting in the GSclk domain, is running or not.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0: Accumulator in PM is disabled</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1: Accumulator in PM is enabled</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td><strong>Render Power Meter Counter Enable</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Render Power Meter Enable :</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Controls whether power meter is running or not.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0: All Power meter counters/accumulators in PM are disabled and held at zero.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1: All power meter counters/accumulators in PM are enabled.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note that this bit does not affect intermediate accumulation/overflow logic elsewhere in the Gfx engine.</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td><strong>Media Power Meter Counter Enable</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Media Power Meter Enable :</td>
</tr>
</tbody>
</table>
# PWRMTRLK - GT_CR_POWER_METER_CTRL

<table>
<thead>
<tr>
<th>2:0</th>
<th>Reserved</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Access:</td>
</tr>
</tbody>
</table>

Controls whether power meter is running or not.

0: All Power meter counters/accumulators in PM are disabled and held at zero.
1: All power meter counters/accumulators in PM are enabled.

Note that this bit does not affect intermediate accumulation/overflow logic elsewhere in the Gfx engine.
## GTFIFOCTL - GTFIFOCTL

### Register Space
MMIO: 0/2/0

### Project
CHV, BSW

### Source
PRM

### Default Value
0x00000000

### Size (in bits)
32

### Address
120008h

### GT FIFO Control

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31</td>
<td><strong>GT_IOSFSB_READ_POLICY</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td>30:16</td>
<td><strong>SPARE15</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 0000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td><strong>GT_FIFO_SB_POLICY</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td><strong>GT_FIFO_BLOB_POLICY</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td><strong>GT_FIFO_PRI_POLICY</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
</tbody>
</table>

- **GT_IOSFSB_READ_POLICY**
  - The SEC must first write this register bit before attempting Gunit register context reads.
  - This is identified usage model requiring IOSF SB reads. Note: For CHV, BSW, Punit has its own IOSF SB P interface.
  - 0 (Default): Abort IOSF SB reads. Only Kf1 and Kf1 status are allowed.
  - 1: Allow IOSF SB reads.

- **GT_FIFO_SB_POLICY**
  - GT.WakeFIFO IOSF SB Policy register
  - 0 (default): If WakeFIFO threshold hit and dedicated IOSF SB buffering is full, stall IOSF SB accesses targeting WakeFIFO.
  - 1: If WakeFIFO threshold is hit AND dedicated IOSF SB buffering is full: a. Drop IOSF SB write requests to WakeFIFO b. IOSF SB reads targeting WakeFIFO return 1s.
  - Evaluated for the request at the head of the IA_WakeFIFO.

- **GT_FIFO_BLOB_POLICY**
  - Default Value: 0b
  - Access: R/W

- **GT_FIFO_PRI_POLICY**
  - Default Value: 0b
  - Access: R/W

*GT_WakeFIFO IOSF Primary Policy register.*
0 (Default): If WakeFIFO threshold hit, stall IOSF Primary accesses targeting WakeFIFO.
### GTFIFOCTL - GTFIFOCTL

**1 : If WakeFIFO threshold is hit:**
- a. Drop IOSF Primary writes to WakeFIFO
- b. IOSF Primary reads targeting WakeFIFO return 1s.
  
  Don’t hang the system, but device 2 may hang.

<table>
<thead>
<tr>
<th>12</th>
<th><strong>Block all Policy</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

BlockALL policy register applies to both HW FIFO and IA GT FIFO. This bit only applies when the GPM BlockALL indication is asserted.

GPM BlockALL indication can assert for: A) CPD and B) for a period of time during RC6 entry.

0 (default): Allow register collapsing and stall request at the head if it is not collapse-able.

1: Stall request at the head.

<table>
<thead>
<tr>
<th>11</th>
<th><strong>RC6_POLICY</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

RC6 policy register applies to both HW FIFO and IA GT FIFO. This bit only applies:
- when BlockALL is not asserted
- within RC6, and
- ONLY when AllowWake register (13_0090h[0]=1).

0 (default): Allow register collapsing and drop request at the head if it is not collapsible.

1: Stall request at the head if within RC6.

Note: All hardware initiated requests should be collapsible.

Note: Starting with Gen8 Gfx, the driver is required to ensure the targeted power well is alive before initiating an access outside shadow register space.

<table>
<thead>
<tr>
<th>10:9</th>
<th><strong>SPARE2</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Default Value: 00b</td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>8</th>
<th><strong>Reserved</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>7:0</th>
<th><strong>SPARE8</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td>Access: RO</td>
</tr>
</tbody>
</table>

**Reserved**

**SPARE2**

**SPARE8**
## GT Function Level Reset Control Message

<table>
<thead>
<tr>
<th>FLRCTLMSG - GT Function Level Reset Control Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 08100h</td>
</tr>
</tbody>
</table>

### GT FLR Control Register

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td><strong>Message Mask</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Message Mask:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>In order to write to bits 15:0, the corresponding message mask bits must be written. For example, for bit 14 to be set, bit 30 needs to be 1: 40004000</td>
</tr>
<tr>
<td>15:1</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved R/W: FLR is not supported on CHV, BSW.</td>
</tr>
</tbody>
</table>
GT INTERRUPT 0 ENABLE REGISTER

<table>
<thead>
<tr>
<th>GT_INTERRUPT0_IER - GT INTERRUPT 0 ENABLE REGISTER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 4430Ch-4430Fh</td>
</tr>
</tbody>
</table>

This table indicates which events are mapped to each bit of the GT Interrupt 0 registers. Bits 15:0 are used for Render CS. Bits 31:16 are used for Blitter CS. The IER enabled Render Interrupt IIR (sticky) bits are ORed together to generate the Render Interrupts Pending bit in the Master Interrupt Control register. The IER enabled Blitter Interrupt IIR (sticky) bits are ORed together to generate the Blitter Interrupts Pending bit in the Master Interrupt Control register.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31</td>
<td>UNUSED0</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>UNUSED1</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>29</td>
<td>UNUSED2</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>UNUSED3</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>27</td>
<td>BCS_WAIT_ON_SEMAPHORE</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BCS wait on semaphore</td>
<td></td>
</tr>
<tr>
<td></td>
<td>26</td>
<td>UNUSED4</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>UNUSED5</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>BCS_CTX_SWITCH_INTERRUPT</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BCS context switch interrupt</td>
<td></td>
</tr>
<tr>
<td></td>
<td>23</td>
<td>UNUSED6</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>Register Name</td>
<td>Access</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>--------------------------------------------------</td>
<td>---------</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>UNUSED7</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>UNUSED8</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>BCS_MI_FLUSH_DWNOTIFY</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td>BCS MI flush DW notify</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>BCS_ERROR_INTERRUPT</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td>BCS error interrupt</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>UNUSED9</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>UNUSED10</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>BCS_MI_USER_INTERRUPT</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td>BCS MI user interrupt</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>UNUSED11</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>UNUSED12</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>UNUSED13</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>UNUSED14</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>CS_WAIT_ON_SEMAPHORE</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CS wait on semaphore</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>CS_L3_COUNTER_SAVE</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CS L3 counter save</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>UNUSED15</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>Register Name</td>
<td>Function Description</td>
<td>Access</td>
<td>Notes</td>
</tr>
<tr>
<td>---------------</td>
<td>----------------------</td>
<td>--------</td>
<td>-------</td>
</tr>
<tr>
<td>GT_INTERRUPT0_IER</td>
<td>GT INTERRUPT 0 ENABLE REGISTER</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>CS_CTX_SWITCH_INTERRUPT</td>
<td>R/W</td>
<td>CS context switch interrupt</td>
</tr>
<tr>
<td>7</td>
<td>PAGE_FAULT_ERROR</td>
<td>R/W</td>
<td>this interrupt is for handling legacy page fault interface for all command streamer (BCS, VCS, RCS, 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 'page fault support' section for more details.</td>
</tr>
<tr>
<td>6</td>
<td>CS_WATCHDOG_COUNTER_EXPIRED</td>
<td>R/W</td>
<td>CS watchdog counter expired</td>
</tr>
<tr>
<td>5</td>
<td>L3PARITYERROR</td>
<td>R/W</td>
<td>L3 parity error</td>
</tr>
<tr>
<td>4</td>
<td>CS_PIPE_CONTROL_NOTIFY</td>
<td>R/W</td>
<td>CS pipe control notify</td>
</tr>
<tr>
<td>3</td>
<td>CS_ERROR_INTERRUPT</td>
<td>R/W</td>
<td>CS error interrupt</td>
</tr>
<tr>
<td>2</td>
<td>UNUSED17</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Reserved</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>CS_MI_USER_INTERRUPT</td>
<td>R/W</td>
<td>CS context switch interrupt</td>
</tr>
</tbody>
</table>
## GT INTERRUPT 0 IDENTITY REGISTER

### GT_INTERRUPT0_IIR - GT INTERRUPT 0 IDENTITY REGISTER

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>44308h-4430Bh</td>
</tr>
</tbody>
</table>

This table indicates which events are mapped to each bit of the GT Interrupt 0 registers. Bits 15:0 are used for Render CS. Bits 31:16 are used for Blitter CS.

The IER enabled Render Interrupt IIR (sticky) bits are ORed together to generate the Render Interrupts Pending bit in the Master Interrupt Control register.

The IER enabled Blitter Interrupt IIR (sticky) bits are ORed together to generate the Blitter Interrupts Pending bit in the Master Interrupt Control register.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31</td>
<td>UNUSED0</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>UNUSED1</td>
</tr>
<tr>
<td></td>
<td>29</td>
<td>UNUSED2</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>UNUSED3</td>
</tr>
<tr>
<td></td>
<td>27</td>
<td>BCS_WAIT_ON_SEMAPHORE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BCS wait on semaphore</td>
</tr>
<tr>
<td></td>
<td>26</td>
<td>UNUSED4</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>UNUSED5</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>BCS_CTX_SWITCH_INTERRUPT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BCS context switch interrupt</td>
</tr>
<tr>
<td></td>
<td>23</td>
<td>UNUSED6</td>
</tr>
<tr>
<td>Register Name</td>
<td>Access</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------------</td>
<td>--------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>GT_INTERRUPT0_IIR - GT INTERRUPT 0 IDENTITY REGISTER</td>
<td></td>
<td>22 UNUSED7</td>
</tr>
<tr>
<td></td>
<td>Access</td>
<td>R/W One Clear</td>
</tr>
<tr>
<td></td>
<td></td>
<td>21 UNUSED8</td>
</tr>
<tr>
<td></td>
<td>Access</td>
<td>R/W One Clear</td>
</tr>
<tr>
<td></td>
<td></td>
<td>20 BCS_MI_FLUSH_DWNOTIFY</td>
</tr>
<tr>
<td></td>
<td>Access</td>
<td>R/W One Clear</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BCS MI flush DW notify</td>
</tr>
<tr>
<td></td>
<td></td>
<td>19 BCS_ERROR_INTERRUPT</td>
</tr>
<tr>
<td></td>
<td>Access</td>
<td>R/W One Clear</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BCS error interrupt</td>
</tr>
<tr>
<td></td>
<td></td>
<td>18 UNUSED9</td>
</tr>
<tr>
<td></td>
<td>Access</td>
<td>R/W One Clear</td>
</tr>
<tr>
<td></td>
<td></td>
<td>17 UNUSED10</td>
</tr>
<tr>
<td></td>
<td>Access</td>
<td>R/W One Clear</td>
</tr>
<tr>
<td></td>
<td></td>
<td>16 BCS_MI_USER_INTERRUPT</td>
</tr>
<tr>
<td></td>
<td>Access</td>
<td>R/W One Clear</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BCS MI user interrupt</td>
</tr>
<tr>
<td></td>
<td></td>
<td>15 UNUSED11</td>
</tr>
<tr>
<td></td>
<td>Access</td>
<td>R/W One Clear</td>
</tr>
<tr>
<td></td>
<td></td>
<td>14 UNUSED12</td>
</tr>
<tr>
<td></td>
<td>Access</td>
<td>R/W One Clear</td>
</tr>
<tr>
<td></td>
<td></td>
<td>13 UNUSED13</td>
</tr>
<tr>
<td></td>
<td>Access</td>
<td>R/W One Clear</td>
</tr>
<tr>
<td></td>
<td></td>
<td>12 UNUSED14</td>
</tr>
<tr>
<td></td>
<td>Access</td>
<td>R/W One Clear</td>
</tr>
<tr>
<td></td>
<td></td>
<td>11 CS_WAIT_ON_SEMAPHORE</td>
</tr>
<tr>
<td></td>
<td>Access</td>
<td>R/W One Clear</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CS wait on semaphore</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10 CS_L3_COUNTER_SAVE</td>
</tr>
<tr>
<td></td>
<td>Access</td>
<td>R/W One Clear</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CS L3 counter save</td>
</tr>
<tr>
<td></td>
<td></td>
<td>9 UNUSED15</td>
</tr>
<tr>
<td></td>
<td>Access</td>
<td>R/W One Clear</td>
</tr>
</tbody>
</table>
## GT_INTERRUPT0_IIR - GT INTERRUPT 0 IDENTITY REGISTER

<table>
<thead>
<tr>
<th>Register</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>8 CS_CTX_SWITCH_INTERRUPT</strong></td>
<td>R/W One Clear</td>
<td>CS context switch interrupt</td>
</tr>
<tr>
<td><strong>7 PAGE_FAULT_ERROR</strong></td>
<td>R/W One Clear</td>
<td>this interrupt is for handling legacy page fault interface for all command streamer (BCS, VCS, RCS, 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 'Page fault support' section for more details.</td>
</tr>
<tr>
<td><strong>6 CS_WATCHDOG_COUNTER_EXPIRED</strong></td>
<td>R/W One Clear</td>
<td>CS watchdog counter expired</td>
</tr>
<tr>
<td><strong>5 L3PARITYERROR</strong></td>
<td>R/W One Clear</td>
<td>L3 parity error</td>
</tr>
<tr>
<td><strong>4 CS_PIPE_CONTROL_NOTIFY</strong></td>
<td>R/W One Clear</td>
<td>CS pipe control notify</td>
</tr>
<tr>
<td><strong>3 CS_ERROR_INTERRUPT</strong></td>
<td>R/W One Clear</td>
<td>CS error interrupt</td>
</tr>
<tr>
<td><strong>2 UNUSED17</strong></td>
<td>R/W One Clear</td>
<td></td>
</tr>
<tr>
<td><strong>1 Reserved</strong></td>
<td>R/W One Clear</td>
<td></td>
</tr>
<tr>
<td><strong>0 CS_MI_USER_INTERRUPT</strong></td>
<td>R/W One Clear</td>
<td>CS context switch interrupt</td>
</tr>
</tbody>
</table>
## GT INTERRUPT 0 MASK REGISTER

### GT_INTERRUPT0_IMR - GT INTERRUPT 0 MASK REGISTER

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31</td>
<td>UNUSED0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td>30</td>
<td></td>
<td>UNUSED1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td>29</td>
<td></td>
<td>UNUSED2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td>28</td>
<td></td>
<td>UNUSED3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td>27</td>
<td></td>
<td>BCS_WAIT_ON_SEMAPHORE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 1b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BCS wait on semaphore</td>
</tr>
<tr>
<td>26</td>
<td></td>
<td>UNUSED4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td>25</td>
<td></td>
<td>UNUSED5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td>24</td>
<td></td>
<td>BCS_CTX_SWITCH_INTERRUPT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 1b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BCS context switch interrupt</td>
</tr>
<tr>
<td>23</td>
<td></td>
<td>UNUSED6</td>
</tr>
</tbody>
</table>

This table indicates which events are mapped to each bit of the GT Interrupt 0 registers. Bits 15:0 are used for Render CS. Bits 31:16 are used for Blitter CS. The IER enabled Render Interrupt IIR (sticky) bits are ORed together to generate the Render Interrupts Pending bit in the Master Interrupt Control register. The IER enabled Blitter Interrupt IIR (sticky) bits are ORed together to generate the Blitter Interrupts Pending bit in the Master Interrupt Control register.
## GT_INTERRUPT0_IMR - GT INTERRUPT 0 MASK REGISTER

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>22 UNUSED7</td>
<td></td>
<td></td>
<td>R/W</td>
</tr>
<tr>
<td>21 UNUSED8</td>
<td></td>
<td></td>
<td>R/W</td>
</tr>
<tr>
<td>20 BCS_MI_FLUSH_DWNOTIFY</td>
<td>BCS MI flush DW notify</td>
<td>1b</td>
<td>R/W</td>
</tr>
<tr>
<td>19 BCS_ERROR_INTERRUPT</td>
<td>BCS error interrupt</td>
<td>1b</td>
<td>R/W</td>
</tr>
<tr>
<td>18 UNUSED9</td>
<td></td>
<td></td>
<td>R/W</td>
</tr>
<tr>
<td>17 UNUSED10</td>
<td></td>
<td></td>
<td>R/W</td>
</tr>
<tr>
<td>16 BCS_MI_USER_INTERRUPT</td>
<td>BCS MI user interrupt</td>
<td>1b</td>
<td>R/W</td>
</tr>
<tr>
<td>15 UNUSED11</td>
<td></td>
<td></td>
<td>R/W</td>
</tr>
<tr>
<td>14 UNUSED12</td>
<td></td>
<td></td>
<td>R/W</td>
</tr>
<tr>
<td>13 UNUSED13</td>
<td></td>
<td></td>
<td>R/W</td>
</tr>
<tr>
<td>12 UNUSED14</td>
<td></td>
<td></td>
<td>R/W</td>
</tr>
<tr>
<td>11 CS_WAIT_ON_SEMAPHORE</td>
<td>CS wait on semaphore</td>
<td>1b</td>
<td>R/W</td>
</tr>
<tr>
<td>Register Name</td>
<td>Default Value</td>
<td>Access</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------------</td>
<td>---------------</td>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>GT_INTERRUPT0_IMR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>10 CS_L3_COUNTER_SAVE</strong></td>
<td>1b</td>
<td>R/W</td>
<td>CS L3 counter save</td>
</tr>
<tr>
<td><strong>9 UNUSED15</strong></td>
<td></td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td><strong>8 CS_CTX_SWITCH_INTERRUPT</strong></td>
<td>1b</td>
<td>R/W</td>
<td>CS context switch interrupt</td>
</tr>
<tr>
<td><strong>7 PAGE_FAULT_ERROR</strong></td>
<td>1b</td>
<td>R/W</td>
<td>this interrupt is for handling legacy page fault interface for all command streamer (BCS, VCS, RCS, 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 'page fault support' section for more details.</td>
</tr>
<tr>
<td><strong>6 CS_WATCHDOG_COUNTER_EXPIRED</strong></td>
<td>1b</td>
<td>R/W</td>
<td>CS watchdog counter expired</td>
</tr>
<tr>
<td><strong>5 L3PARITYERROR</strong></td>
<td>1b</td>
<td>R/W</td>
<td>L3 parity error</td>
</tr>
<tr>
<td><strong>4 CS_PIPE_CONTROL_NOTIFY</strong></td>
<td>1b</td>
<td>R/W</td>
<td>CS pipe control notify</td>
</tr>
<tr>
<td><strong>3 CS_ERROR_INTERRUPT</strong></td>
<td>1b</td>
<td>R/W</td>
<td>CS error interrupt</td>
</tr>
</tbody>
</table>
## GT_INTERRUPT0_IMR - GT INTERRUPT 0 MASK REGISTER

<table>
<thead>
<tr>
<th>Offset</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>UNUSED17</td>
<td>0</td>
<td>R/W</td>
</tr>
<tr>
<td>1</td>
<td>Reserved</td>
<td></td>
<td>R/W</td>
</tr>
<tr>
<td>0</td>
<td>CS_MI_USER_INTERRUPT</td>
<td>1b</td>
<td>R/W</td>
</tr>
</tbody>
</table>

- **Access**: R/W
- **Default Value**: 1b
- **CS context switch interrupt**
## GT INTERRUPT 0 STATUS REGISTER

<table>
<thead>
<tr>
<th>GT_INTERRUPT0_ISR</th>
<th>GT INTERRUPT 0 STATUS REGISTER</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space</strong></td>
<td>MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project</strong></td>
<td>CHV, BSW</td>
</tr>
<tr>
<td><strong>Source</strong></td>
<td>PRM</td>
</tr>
<tr>
<td><strong>Default Value</strong></td>
<td>0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits)</strong></td>
<td>32</td>
</tr>
<tr>
<td><strong>Address</strong></td>
<td>44300h-44303h</td>
</tr>
</tbody>
</table>

This table indicates which events are mapped to each bit of the GT Interrupt 0 registers. Bits 15:0 are used for Render CS. Bits 31:16 are used for Blitter CS. The IER enabled Render Interrupt IIR (sticky) bits are ORed together to generate the Render Interrupts Pending bit in the Master Interrupt Control register. The IER enabled Blitter Interrupt IIR (sticky) bits are ORed together to generate the Blitter Interrupts Pending bit in the Master Interrupt Control register.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31</td>
<td><strong>UNUSED0</strong></td>
<td>RO</td>
</tr>
<tr>
<td>30</td>
<td><strong>UNUSED1</strong></td>
<td>RO</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td><strong>UNUSED2</strong></td>
<td>RO</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td><strong>UNUSED3</strong></td>
<td>RO</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td><strong>BCS_WAIT_ON_SEMAPHORE</strong></td>
<td>RO</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>BCS wait on semaphore</td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td><strong>UNUSED4</strong></td>
<td>RO</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td><strong>UNUSED5</strong></td>
<td>RO</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td><strong>BCS_CTX_SWITCH_INTERRUPT</strong></td>
<td>RO</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>BCS context switch interrupt</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td><strong>UNUSED6</strong></td>
<td>RO</td>
<td></td>
</tr>
</tbody>
</table>

*Doc Ref # IHD-OS-CHV-BSW-Vol 2c- 10.15*
# GT INTERRUPT0_ISR - GT INTERRUPT 0 STATUS REGISTER

<table>
<thead>
<tr>
<th>Row</th>
<th>Register Name</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>22</td>
<td>UNUSED7</td>
<td>RO</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>UNUSED8</td>
<td>RO</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>BCS_MI_FLUSH_DWNOTIFY</td>
<td>RO</td>
<td>BCS MI flush DW notify</td>
</tr>
<tr>
<td>19</td>
<td>BCS_ERROR_INTERRUPT</td>
<td>RO</td>
<td>BCS error interrupt</td>
</tr>
<tr>
<td>18</td>
<td>UNUSED9</td>
<td>RO</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>UNUSED10</td>
<td>RO</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>BCS_MI_USER_INTERRUPT</td>
<td>RO</td>
<td>BCS MI user interrupt</td>
</tr>
<tr>
<td>15</td>
<td>UNUSED11</td>
<td>RO</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>UNUSED12</td>
<td>RO</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>UNUSED13</td>
<td>RO</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>UNUSED14</td>
<td>RO</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>CS_WAIT_ON_SEMAPHORE</td>
<td>RO</td>
<td>CS wait on semaphore</td>
</tr>
<tr>
<td>10</td>
<td>CS_L3_COUNTER_SAVE</td>
<td>RO</td>
<td>CS L3 counter save</td>
</tr>
<tr>
<td>9</td>
<td>UNUSED15</td>
<td>RO</td>
<td></td>
</tr>
<tr>
<td>Register Name</td>
<td>Description</td>
<td>Access</td>
<td></td>
</tr>
<tr>
<td>--------------------------------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>--------</td>
<td></td>
</tr>
<tr>
<td>CS_CTX_SWITCH_INTERRUPT</td>
<td>CS context switch interrupt</td>
<td>RO</td>
<td></td>
</tr>
<tr>
<td>PAGE_FAULT_ERROR</td>
<td>this interrupt is for handling legacy page fault interface for all command streamer (BCS, VCS, RCS, 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 'page fault support' section for more details.</td>
<td>RO</td>
<td></td>
</tr>
<tr>
<td>CS_WATCHDOG_COUNTER_EXPIRED</td>
<td>CS watchdog counter expired</td>
<td>RO</td>
<td></td>
</tr>
<tr>
<td>L3PARITYERROR</td>
<td>L3 parity error</td>
<td>RO</td>
<td></td>
</tr>
<tr>
<td>CS_PIPE_CONTROL_NOTIFY</td>
<td>CS pipe control notify</td>
<td>RO</td>
<td></td>
</tr>
<tr>
<td>CS_ERROR_INTERRUPT</td>
<td>CS error interrupt</td>
<td>RO</td>
<td></td>
</tr>
<tr>
<td>UNUSED17</td>
<td></td>
<td>RO</td>
<td></td>
</tr>
<tr>
<td>Reserved</td>
<td></td>
<td>RO</td>
<td></td>
</tr>
<tr>
<td>CS_MI_USER_INTERRUPT</td>
<td>CS context switch interrupt</td>
<td>RO</td>
<td></td>
</tr>
</tbody>
</table>
## GT INTERRUPT1 ENABLE REGISTER

<table>
<thead>
<tr>
<th>GT_INTERRUPT1_IER - GT INTERRUPT1 ENABLE REGISTER</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 4431Ch-4431Fh</td>
</tr>
</tbody>
</table>

This table indicates which events are mapped to each bit of the GT Interrupt 0 registers. Bits 15:0 are used for VCS1. Bits 31:16 are used for VCS2.

The IER enabled VCS1 Interrupt IIR (sticky) bits are ORed together to generate the VCS1 Interrupts Pending bit in the Master Interrupt Control register.

The IER enabled VCS2 Interrupt IIR (sticky) bits are ORed together to generate the VCS2 Interrupt Pending bit in the Master Interrupt Control register.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31</td>
<td>UNUSED0</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>UNUSED1</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>29</td>
<td>UNUSED2</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>UNUSED3</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>27</td>
<td>VCS2_WAIT_ON_SEMAPHORE</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VCS2 wait on semaphore</td>
<td></td>
</tr>
<tr>
<td></td>
<td>26</td>
<td>UNUSED4</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>Reserved</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>VCS2_CTX_SWITCH_INTERRUPT</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VCS2 context switch interrupt</td>
<td></td>
</tr>
<tr>
<td></td>
<td>23</td>
<td>UNUSED5</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>22</td>
<td>VCS2_WATCHDOG_COUNTER_EXPIRED</td>
<td>R/W</td>
</tr>
</tbody>
</table>
# GT_INTERRUPT1_IER - GT INTERRUPT1 ENABLE REGISTER

<table>
<thead>
<tr>
<th>Offset</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>21</td>
<td>Reserved</td>
<td>R/W</td>
</tr>
<tr>
<td>20</td>
<td>VCS2_MI.Flush_DWNotify</td>
<td>R/W</td>
</tr>
<tr>
<td>19</td>
<td>VCS2.Error.Interrupt</td>
<td>R/W</td>
</tr>
<tr>
<td>18</td>
<td>Unused6</td>
<td>R/W</td>
</tr>
<tr>
<td>17</td>
<td>Unused7</td>
<td>R/W</td>
</tr>
<tr>
<td>16</td>
<td>VCS2_MI.User.Interrupt</td>
<td>R/W</td>
</tr>
<tr>
<td>15</td>
<td>Unused8</td>
<td>R/W</td>
</tr>
<tr>
<td>14</td>
<td>Unused9</td>
<td>R/W</td>
</tr>
<tr>
<td>13</td>
<td>Unused10</td>
<td>R/W</td>
</tr>
<tr>
<td>12</td>
<td>Unused11</td>
<td>R/W</td>
</tr>
<tr>
<td>11</td>
<td>VCS1.Wait_on_Semaphore</td>
<td>R/W</td>
</tr>
<tr>
<td>10</td>
<td>Unused12</td>
<td>R/W</td>
</tr>
<tr>
<td>9</td>
<td>Reserved</td>
<td>R/W</td>
</tr>
<tr>
<td>8</td>
<td>VCS1_CTX.Switch.Interrupt</td>
<td>R/W</td>
</tr>
</tbody>
</table>

- **Access:** R/W
- **VCS2 watchdog counter expired**
- **VCS2 MI flush DW notify**
- **VCS2 error interrupt**
- **VCS2 MI user interrupt**
- **VCS1 wait on semaphore**
- **VCS1 context switch interrupt**
<table>
<thead>
<tr>
<th>Register Description</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GT_INTERRUPT1_IER - GT INTERRUPT1 ENABLE REGISTER</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7 Unused13</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>6 <strong>VCS1_WATCHDOG_COUNTER_EXPIRED</strong></td>
<td>R/W</td>
<td>VCS1 watchdog counter expired</td>
</tr>
<tr>
<td>5 <strong>Reserved</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 <strong>VCS1_MI_FLUSH_DWNOTIFY</strong></td>
<td>R/W</td>
<td>VCS1 MI flush DW notify</td>
</tr>
<tr>
<td>3 <strong>VCS1_ERROR_INTERRUPT</strong></td>
<td>R/W</td>
<td>VCS1 error interrupt</td>
</tr>
<tr>
<td>2 Unused14</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>1 Unused15</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>0 <strong>VCS1_MI_USER_INTERRUPT</strong></td>
<td>R/W</td>
<td>VCS1 MI user interrupt</td>
</tr>
</tbody>
</table>
## GT INTERRUPT1 IDENTITY REGISTER

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31</td>
<td>UNUSED0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W One Clear</td>
</tr>
<tr>
<td>0</td>
<td>30</td>
<td>UNUSED1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W One Clear</td>
</tr>
<tr>
<td>0</td>
<td>29</td>
<td>UNUSED2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W One Clear</td>
</tr>
<tr>
<td>0</td>
<td>28</td>
<td>UNUSED3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W One Clear</td>
</tr>
<tr>
<td>0</td>
<td>27</td>
<td>VCS2_WAIT_ON_SEMAPHORE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W One Clear</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VCS2 wait on semaphore</td>
</tr>
<tr>
<td>0</td>
<td>26</td>
<td>UNUSED4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W One Clear</td>
</tr>
<tr>
<td>0</td>
<td>25</td>
<td>Reserved</td>
</tr>
<tr>
<td>0</td>
<td>24</td>
<td>VCS2_CTX_SWITCH_INTERRUPT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W One Clear</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VCS2 context switch interrupt</td>
</tr>
<tr>
<td>0</td>
<td>23</td>
<td>UNUSED5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W One Clear</td>
</tr>
<tr>
<td>0</td>
<td>22</td>
<td>VCS2_WATCHDOG_COUNTER_EXPIRED</td>
</tr>
<tr>
<td>Register Description</td>
<td>Access</td>
<td></td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>--------------</td>
<td></td>
</tr>
<tr>
<td>GT_INTERRUPT1_IIR - GT INTERRUPT1 IDENTITY REGISTER</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21  GT INTERRUPT1.Identity Register</td>
<td>R/W One Clear</td>
<td></td>
</tr>
<tr>
<td>20  VCS2 watchdog counter expired</td>
<td>R/W One Clear</td>
<td></td>
</tr>
<tr>
<td>19  VCS2 MI flush DW notify</td>
<td>R/W One Clear</td>
<td></td>
</tr>
<tr>
<td>18  VCS2 error interrupt</td>
<td>R/W One Clear</td>
<td></td>
</tr>
<tr>
<td>17  UNUSED6</td>
<td>R/W One Clear</td>
<td></td>
</tr>
<tr>
<td>16  VCS2 MI user interrupt</td>
<td>R/W One Clear</td>
<td></td>
</tr>
<tr>
<td>15  UNUSED8</td>
<td>R/W One Clear</td>
<td></td>
</tr>
<tr>
<td>14  UNUSED9</td>
<td>R/W One Clear</td>
<td></td>
</tr>
<tr>
<td>13  UNUSED10</td>
<td>R/W One Clear</td>
<td></td>
</tr>
<tr>
<td>12  UNUSED11</td>
<td>R/W One Clear</td>
<td></td>
</tr>
<tr>
<td>11  VCS1 wait on semaphore</td>
<td>R/W One Clear</td>
<td></td>
</tr>
<tr>
<td>10  UNUSED12</td>
<td>R/W One Clear</td>
<td></td>
</tr>
<tr>
<td>9   Reserved</td>
<td>R/W One Clear</td>
<td></td>
</tr>
<tr>
<td>8   VCS1 context switch interrupt</td>
<td>R/W One Clear</td>
<td></td>
</tr>
<tr>
<td>Bit</td>
<td>Description</td>
<td>Access</td>
</tr>
<tr>
<td>-----</td>
<td>--------------------------------------------------</td>
<td>------------</td>
</tr>
<tr>
<td>7</td>
<td><strong>UNUSED13</strong></td>
<td>R/W One Clear</td>
</tr>
<tr>
<td>6</td>
<td><strong>VCS1_WATCHDOG_COUNTER_EXPIRED</strong></td>
<td>R/W One Clear</td>
</tr>
<tr>
<td></td>
<td>VCS1 watchdog counter expired</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td><strong>Reserved</strong></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td><strong>VCS1_MI_FLUSH_DWNOTIFY</strong></td>
<td>R/W One Clear</td>
</tr>
<tr>
<td></td>
<td>VCS1 MI flush DW notify</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td><strong>VCS1_ERROR_INTERRUPT</strong></td>
<td>R/W One Clear</td>
</tr>
<tr>
<td></td>
<td>VCS1 error interrupt</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td><strong>UNUSED14</strong></td>
<td>R/W One Clear</td>
</tr>
<tr>
<td>1</td>
<td><strong>UNUSED15</strong></td>
<td>R/W One Clear</td>
</tr>
<tr>
<td>0</td>
<td><strong>VCS1_MI_USER_INTERRUPT</strong></td>
<td>R/W One Clear</td>
</tr>
<tr>
<td></td>
<td>VCS1 MI user interrupt</td>
<td></td>
</tr>
</tbody>
</table>
## GT INTERRUPT1 MASK REGISTER

### GT_INTERRUPT1_IMR - GT INTERRUPT1 MASK REGISTER

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31</td>
<td>UNUSED0</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>UNUSED1</td>
</tr>
<tr>
<td></td>
<td>29</td>
<td>UNUSED2</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>UNUSED3</td>
</tr>
<tr>
<td></td>
<td>27</td>
<td>VCS2_WAIT_ON_SEMAPHORE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>26</td>
<td>UNUSED4</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>VCS2_CTX_SWITCH_INTERRUPT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>23</td>
<td>UNUSED5</td>
</tr>
</tbody>
</table>

This table indicates which events are mapped to each bit of the GT Interrupt 0 registers. Bits 15:0 are used for VCS1. Bits 31:16 are used for VCS2.

The IER enabled VCS1 Interrupt IIR (sticky) bits are ORed together to generate the VCS1 Interrupts Pending bit in the Master Interrupt Control register.

The IER enabled VCS2 Interrupt IIR (sticky) bits are ORed together to generate the VCS2 Interrupt Pending bit in the Master Interrupt Control register.
<table>
<thead>
<tr>
<th>GT_INTERRUPT1_IMR - GT INTERRUPT1 MASK REGISTER</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>22</strong></td>
</tr>
<tr>
<td>Default Value:</td>
</tr>
<tr>
<td>Access:</td>
</tr>
<tr>
<td>VCS2 watchdog counter expired</td>
</tr>
<tr>
<td><strong>21</strong></td>
</tr>
<tr>
<td><strong>20</strong></td>
</tr>
<tr>
<td>Default Value:</td>
</tr>
<tr>
<td>Access:</td>
</tr>
<tr>
<td>VCS2 MI flush DW notify</td>
</tr>
<tr>
<td><strong>19</strong></td>
</tr>
<tr>
<td>Default Value:</td>
</tr>
<tr>
<td>Access:</td>
</tr>
<tr>
<td>VCS2 error interrupt</td>
</tr>
<tr>
<td><strong>18</strong></td>
</tr>
<tr>
<td>Access:</td>
</tr>
<tr>
<td><strong>17</strong></td>
</tr>
<tr>
<td>Access:</td>
</tr>
<tr>
<td><strong>16</strong></td>
</tr>
<tr>
<td>Default Value:</td>
</tr>
<tr>
<td>Access:</td>
</tr>
<tr>
<td>VCS2 MI user interrupt</td>
</tr>
<tr>
<td><strong>15</strong></td>
</tr>
<tr>
<td>Access:</td>
</tr>
<tr>
<td><strong>14</strong></td>
</tr>
<tr>
<td>Access:</td>
</tr>
<tr>
<td><strong>13</strong></td>
</tr>
<tr>
<td>Access:</td>
</tr>
<tr>
<td><strong>12</strong></td>
</tr>
<tr>
<td>Access:</td>
</tr>
<tr>
<td><strong>11</strong></td>
</tr>
<tr>
<td>Default Value:</td>
</tr>
<tr>
<td>Access:</td>
</tr>
<tr>
<td>VCS1 wait on semaphore</td>
</tr>
</tbody>
</table>
## GT_INTERRUPT1_IMR - GT INTERRUPT1 MASK REGISTER

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>UNUSED12</td>
<td>R/W</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Reserved</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>VCS1_CTX_SWITCH_INTERRUPT</td>
<td>1b</td>
<td>R/W</td>
<td>VCS1 context switch interrupt</td>
</tr>
<tr>
<td>7</td>
<td>UNUSED13</td>
<td>R/W</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>VCS1_WATCHDOG_COUNTER_EXPIRED</td>
<td>1b</td>
<td>R/W</td>
<td>VCS1 watchdog counter expired</td>
</tr>
<tr>
<td>5</td>
<td>Reserved</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>VCS1_MI_FLUSH_DWNotify</td>
<td>1b</td>
<td>R/W</td>
<td>VCS1 MI flush DW notify</td>
</tr>
<tr>
<td>3</td>
<td>VCS1_ERROR_INTERRUPT</td>
<td>1b</td>
<td>R/W</td>
<td>VCS1 error interrupt</td>
</tr>
<tr>
<td>2</td>
<td>UNUSED14</td>
<td>R/W</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>UNUSED15</td>
<td>R/W</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>VCS1_MI_USER_INTERRUPT</td>
<td>1b</td>
<td>R/W</td>
<td>VCS1 MI user interrupt</td>
</tr>
</tbody>
</table>
GT INTERRUPT1 STATUS REGISTER

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31</td>
<td>UNUSED0</td>
<td>RO</td>
</tr>
<tr>
<td>30</td>
<td>30</td>
<td>UNUSED1</td>
<td>RO</td>
</tr>
<tr>
<td>29</td>
<td>29</td>
<td>UNUSED2</td>
<td>RO</td>
</tr>
<tr>
<td>28</td>
<td>28</td>
<td>UNUSED3</td>
<td>RO</td>
</tr>
<tr>
<td>27</td>
<td>27</td>
<td>VCS2_WAIT_ON_SEMAPHORE</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VCS2 wait on semaphore</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>26</td>
<td>UNUSED4</td>
<td>RO</td>
</tr>
<tr>
<td>25</td>
<td>25</td>
<td>Reserved</td>
<td>RO</td>
</tr>
<tr>
<td>24</td>
<td>24</td>
<td>VCS2_CTX_SWITCH_INTERRUPT</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VCS2 context switch interrupt</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>23</td>
<td>UNUSED5</td>
<td>RO</td>
</tr>
<tr>
<td>22</td>
<td>22</td>
<td>VCS2_WATCHDOG_COUNTER_EXPIRED</td>
<td>RO</td>
</tr>
<tr>
<td>Address</td>
<td>Access</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>--------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>0x0000</td>
<td>RO</td>
<td>VCS2 watchdog counter expired</td>
<td></td>
</tr>
<tr>
<td>0x0001</td>
<td>RO</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>0x0002</td>
<td>RO</td>
<td>VCS2_MI_FLUSH_DWNOTIFY</td>
<td></td>
</tr>
<tr>
<td>0x0003</td>
<td>RO</td>
<td>VCS2 error interrupt</td>
<td></td>
</tr>
<tr>
<td>0x0004</td>
<td>RO</td>
<td>Unused6</td>
<td></td>
</tr>
<tr>
<td>0x0005</td>
<td>RO</td>
<td>Unused7</td>
<td></td>
</tr>
<tr>
<td>0x0006</td>
<td>RO</td>
<td>VCS2_MI_USER_INTERRUPT</td>
<td></td>
</tr>
<tr>
<td>0x0007</td>
<td>RO</td>
<td>Unused8</td>
<td></td>
</tr>
<tr>
<td>0x0008</td>
<td>RO</td>
<td>Unused9</td>
<td></td>
</tr>
<tr>
<td>0x0009</td>
<td>RO</td>
<td>Unused10</td>
<td></td>
</tr>
<tr>
<td>0x000A</td>
<td>RO</td>
<td>Unused11</td>
<td></td>
</tr>
<tr>
<td>0x000B</td>
<td>RO</td>
<td>VCS1_WAIT_ON_SEMAPHORE</td>
<td></td>
</tr>
<tr>
<td>0x000C</td>
<td>RO</td>
<td>Unused12</td>
<td></td>
</tr>
<tr>
<td>0x000D</td>
<td>RO</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>0x000E</td>
<td>RO</td>
<td>VCS1_CTX_SWITCH_INTERRUPT</td>
<td></td>
</tr>
<tr>
<td>Bit</td>
<td>Field Name</td>
<td>Description</td>
<td>Access</td>
</tr>
<tr>
<td>-----</td>
<td>------------------------------------</td>
<td>----------------------------------------------</td>
<td>--------</td>
</tr>
<tr>
<td>7</td>
<td>UNUSED13</td>
<td>Unused</td>
<td>RO</td>
</tr>
<tr>
<td>6</td>
<td>VCS1_WATCHDOG_COUNTER_EXPIRED</td>
<td>VCS1 watchdog counter expired</td>
<td>RO</td>
</tr>
<tr>
<td>5</td>
<td>Reserved</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>VCS1_MI_FLUSH_DWNOTIFY</td>
<td>VCS1 MI flush DW notify</td>
<td>RO</td>
</tr>
<tr>
<td>3</td>
<td>VCS1_ERROR_INTERRUPT</td>
<td>VCS1 error interrupt</td>
<td>RO</td>
</tr>
<tr>
<td>2</td>
<td>UNUSED14</td>
<td>Unused</td>
<td>RO</td>
</tr>
<tr>
<td>1</td>
<td>UNUSED15</td>
<td>Unused</td>
<td>RO</td>
</tr>
<tr>
<td>0</td>
<td>VCS1_MI_USER_INTERRUPT</td>
<td>VCS1 MI user interrupt</td>
<td>RO</td>
</tr>
</tbody>
</table>
## GT INTERRUPT3 ENABLE REGISTER

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:29</td>
<td><strong>UNUSED0</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td><strong>Reserved</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>27:17</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td></td>
<td>15:12</td>
<td><strong>UNUSED2</strong></td>
<td>R/W</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td><strong>VECS_WAIT_SEMAPHORE</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VECS wait on semaphore</td>
<td></td>
</tr>
<tr>
<td>10:9</td>
<td></td>
<td><strong>UNUSED3</strong></td>
<td>R/W</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td><strong>VECS_CTX_SWITCH_INT</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VECS context switch interrupt</td>
<td></td>
</tr>
<tr>
<td>7:5</td>
<td></td>
<td><strong>UNUSED4</strong></td>
<td>R/W</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td><strong>VECS_MI_FLUSH_DWNOTIFY</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VECS MI Flush DW Notify</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td><strong>VECS_ERR_INT</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VECS error interrupt</td>
<td></td>
</tr>
</tbody>
</table>

This table indicates which events are mapped to each bit of the GT interrupt 3 register. The VEDBOX Interrupt IIR sticky bits are Or’ed together to generate VEDBOX interrupt pending bit in the master interrupt control register.
## GT_INTERRUPT3_IER - GT INTERRUPT3 ENABLE REGISTER

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>2:1</td>
<td>UNUSED5</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>VECS_MU_USER_INT</td>
<td>R/W</td>
<td>VECS MI user interrupt</td>
</tr>
</tbody>
</table>
# GT INTERRUPT3 IDENTITY REGISTER

<table>
<thead>
<tr>
<th>GT_INTERRUPT3_IIR - GT INTERRUPT3 IDENTITY REGISTER</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 44338h-4433Bh</td>
</tr>
</tbody>
</table>

This table indicates which events are mapped to each bit of the GT interrupt 3 register. The VEDBOX Interrupt IIR sticky bits are Or-ed together to generate VEDBOX interrupt pending bit in the master interrupt control register.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:29</td>
<td><strong>UNUSED0</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W One Clear</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td>27:17</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td>16</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td>15:12</td>
<td><strong>UNUSED2</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W One Clear</td>
</tr>
<tr>
<td>1</td>
<td>11</td>
<td><strong>VECS_WAIT_SEMAPHORE</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W One Clear</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VECS wait on semaphore</td>
</tr>
<tr>
<td>10:9</td>
<td></td>
<td><strong>UNUSED3</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W One Clear</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td><strong>VECS_CTX_SWITCH_INT</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W One Clear</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VECS context switch interrupt</td>
</tr>
<tr>
<td>7:5</td>
<td></td>
<td><strong>UNUSED4</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W One Clear</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td><strong>VECS_MI_FLUSH_DWNOTIFY</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W One Clear</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VECS MI Flush DW Notify</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td><strong>VECS_ERR_INT</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W One Clear</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VECS error interrupt</td>
</tr>
<tr>
<td>Address</td>
<td>Field</td>
<td>Access</td>
</tr>
<tr>
<td>---------</td>
<td>-----------------</td>
<td>-------------</td>
</tr>
<tr>
<td>2:1</td>
<td><strong>UNUSED5</strong></td>
<td>R/W One Clear</td>
</tr>
<tr>
<td>0</td>
<td><strong>VECS_MI_USER_INT</strong></td>
<td>R/W One Clear</td>
</tr>
</tbody>
</table>
## GT INTERRUPT3 MASK REGISTER

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:29</td>
<td>UNUSED0</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>Reserved</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td>27:17</td>
<td>Reserved</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>10:9</td>
<td>15:12</td>
<td>UNUSED2</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>11</td>
<td>VECS_WAIT_SEMAPHORE</td>
<td>R/W</td>
<td>1b</td>
</tr>
<tr>
<td>7:5</td>
<td>8</td>
<td>VECS_CTX_SWITCH_INT</td>
<td>R/W</td>
<td>1b</td>
</tr>
<tr>
<td>4</td>
<td>7:5</td>
<td>UNUSED4</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>VECS_MI_FLUSH_DWNOTIFY</td>
<td>R/W</td>
<td>1b</td>
</tr>
</tbody>
</table>
### GT_INTERRUPT3_IMR - GT INTERRUPT3 MASK REGISTER

<table>
<thead>
<tr>
<th>Offset</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>VECS_ERR_INT</td>
<td>1b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>VECS error interrupt</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2:1</td>
<td>UNUSED5</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>VECS_MI_USER_INT</td>
<td>1b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>VECS MI user interrupt</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# GT INTERRUPT3 STATUS REGISTER

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:29</td>
<td>UNUSED0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>28</td>
<td>Reserved</td>
<td>Reserved</td>
</tr>
<tr>
<td>27:17</td>
<td>Reserved</td>
<td>Reserved</td>
</tr>
<tr>
<td>16</td>
<td>Reserved</td>
<td>Reserved</td>
</tr>
<tr>
<td>15:12</td>
<td>UNUSED2</td>
<td>Access: RO</td>
</tr>
<tr>
<td>11</td>
<td>VECS_WAIT_SEMAPHORE</td>
<td>Access: RO</td>
</tr>
<tr>
<td>10:9</td>
<td>UNUSED3</td>
<td>Access: RO</td>
</tr>
<tr>
<td>8</td>
<td>VECS_CTX_SWITCH_INT</td>
<td>Access: RO</td>
</tr>
<tr>
<td>7:5</td>
<td>UNUSED4</td>
<td>Access: RO</td>
</tr>
<tr>
<td>4</td>
<td>VECS_MI_FLUSH_DWNOTIFY</td>
<td>Access: RO</td>
</tr>
<tr>
<td>3</td>
<td>VECS_ERR_INT</td>
<td>Access: RO</td>
</tr>
<tr>
<td>Address</td>
<td>Description</td>
<td>Access</td>
</tr>
<tr>
<td>---------</td>
<td>------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>2:1</td>
<td>UNUSED5</td>
<td>RO</td>
</tr>
<tr>
<td>0</td>
<td>VECS_MI_USER_INT</td>
<td>RO</td>
</tr>
</tbody>
</table>

- **UNUSED5**: Access: RO
- **VECS_MI_USER_INT**: Access: RO

VECS MI user interrupt
**GTLC_PW_STAT**

**Register Space:** MMIO: 0/2/0  
**Project:** CHV, BSW  
**Source:** PRM  
**Default Value:** 0x00000000  
**Size (in bits):** 32  
**Address:** 130094h

This register contains Ack and status information for power well requests.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:8</td>
<td><strong>RESERVED</strong></td>
<td>000000h</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Reserved</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td><strong>Reserved</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td><strong>Reserved</strong></td>
<td>0b</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Reserved</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td><strong>Reserved</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td><strong>Reserved</strong></td>
<td>0b</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Reserved</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3:2</td>
<td></td>
<td><strong>Reserved</strong></td>
<td>00b</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Reserved</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td><strong>ALLOWWAKEERR</strong></td>
<td>0b</td>
<td>R/W One Clear</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HW set, SW cleared. When access to media or render is observed when ALLOWAKE=0, the ALLOWWAKEERR bit will be set. It will be up to SW or a power cycle to clear the ALLOWWAKEERR bit.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td><strong>ALLOWWAKEACK</strong></td>
<td>0b</td>
<td></td>
</tr>
<tr>
<td><strong>GTLC_PW_STAT - GTLC_PW_STAT</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Access:</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Indicates that the allow wake request has been completed.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>RO</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**GTLC_SURVIVE - GTLC_SURVIVE**

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:12</td>
<td><strong>RESERVED</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td><strong>gvd_pmu_bonus1</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>gvd_pmu_bonus1_zczfwoh</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td><strong>gvd_pmu_bonus0</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>gvd_pmu_bonus0_zczfwoh</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td><strong>pmu_gvd_bonus1</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>pmu_gvd_bonus1_zczfwoh</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td><strong>pmu_gvd_bonus0</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>pmu_gvd_bonus0_zczfwoh</td>
</tr>
<tr>
<td>7:4</td>
<td></td>
<td><strong>RESERVED</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
</tbody>
</table>

This register contains configuration bits for modes that the PM unit should operate in.

Register Space: MMIO: 0/2/0
Project: CHV, BSW
Source: PRM
Default Value: 0x00000000
Size (in bits): 32
Address: 130098h
### GFXCLKSTATUS

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>GFXCLKSTATUS</td>
<td>0b</td>
<td>RO</td>
</tr>
</tbody>
</table>

This bit is used as a way to confirm that the GFX clocks have been turned on with bit 2. SW would normally write a one to bit2 and then poll on this bit until.

### GFXCLKFORCEON

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>GFXCLKFORCEON</td>
<td>0b</td>
<td>R/W</td>
</tr>
</tbody>
</table>

When this bit is set to a '1, the gvd_pmu_gfxclockstartreq_zczfwoh signal to the Punit will be forced to a one.

### SPAREBIT

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SPAREBIT</td>
<td>0b</td>
<td>R/W</td>
</tr>
</tbody>
</table>

Description

Spare bit for CHV, BSW.
Project

CHV, BSW

### GangMediaRender

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>GangMediaRender</td>
<td>0b</td>
<td>R/W</td>
</tr>
</tbody>
</table>

0 : The media and render power wells are brought down and up independently.
1 : PM unit will treat both wells as a unit, taking well down together and up together.
# GTLC_WAKE

<table>
<thead>
<tr>
<th><strong>GTLC_WAKE - GTLC_WAKE</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 130090h</td>
</tr>
</tbody>
</table>

GT Wake Control.
This register is used as a way for the Punit to control the GTLC Render and Media Power wells.
Writing bit 0 during normal operation may result in a hang.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:26</td>
<td><strong>RESERVED</strong></td>
</tr>
<tr>
<td></td>
<td>31:26</td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td>31:26</td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td>25</td>
<td></td>
<td><strong>RenderContextExists</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The usage is for support of s0ix, where Punit/Driver would restore the value in this bit no later than the time a ‘1 is written to ALLOWWAKEREQ (bit 0 of this register) during s0ix exit. SW should not write ‘1 to this bit during a cold boot exit or warm reset exit. CZ reset is the only thing that can clear this bit. This bit is set by HW after the first Render context save after cold boot or warm reset. SW can only write a one to this bit. If SW tries to write a zero, the previous value will be maintained.</td>
</tr>
<tr>
<td>24</td>
<td></td>
<td><strong>MediaContextExists</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The usage is for support of s0ix, where Punit/Driver would restore the value in this bit no later than the time a ‘1 is written to ALLOWWAKEREQ (bit 0 of this register) during s0ix exit. SW should not write ‘1 to this bit during a cold boot exit or warm reset exit. CZ reset is the only thing that can clear this bit. This bit is set by HW after the first Media context save after cold boot or warm reset. SW can only write a one to this bit. If SW tries to write a zero, the previous value will be maintained.</td>
</tr>
<tr>
<td>23:8</td>
<td></td>
<td><strong>Common No-wake Hysteresis timer for gfxstartclkreq</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0000h</td>
</tr>
</tbody>
</table>
GTLC_WAKE - GTLC_WAKE

<table>
<thead>
<tr>
<th>Access:</th>
<th>R/W</th>
</tr>
</thead>
<tbody>
<tr>
<td>New for CHV, BSW. Hysteresis timer would be used to deassert gfxclockstartreq in case of both wells being in standby and common no-wake request occurs. Counter will be incremented on 60ns interval.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>7:1</th>
<th>RESERVED</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Value:</td>
<td>00h</td>
</tr>
<tr>
<td>Access:</td>
<td>RO</td>
</tr>
<tr>
<td>Reserved</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>0</th>
<th>ALLOWWAKEREQ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Value:</td>
<td>0b</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>This bit is used as a way for the driver to make sure GTLC Render AND MEDIA engines do not wake while powered down. The usage is specifically with s0ix, where the driver wants to access the message channel common well (GScik domain) and does not want the render/media wells to wake up. To remove ambiguity, this bit should be set to ‘1 and the ALLOWWAKEACK should be observed to the ‘1 before the FWAKEMEDIAREQ/FWAKERENDERREQ are set. Care must be taken to ONLY use this register prior to Gunit being powered down for S0ix or after registers have been initialized.</td>
<td></td>
</tr>
</tbody>
</table>
### GT Mode Register

<table>
<thead>
<tr>
<th>GT_MODE - GT Mode Register</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> RenderCS</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000 CHV, BSW</td>
</tr>
<tr>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Trusted Type:</strong> 1</td>
</tr>
<tr>
<td><strong>Address:</strong> 07008h</td>
</tr>
<tr>
<td><strong>Valid Projects:</strong> CHV, BSW</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td>Mask</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: WO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: Mask[15:0]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Must be set to modify corresponding bit in Bits 15:0. (All implemented bits)</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>EU Local Thread Checking Enable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: r/w</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This field configures the EU local thread checking. If enable the stateless access will be checked against the local thread's scratch space size and start address.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
<th>Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>Disable [Default]</td>
<td>EU local thread checking is disabled.</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>1h</td>
<td>Enable</td>
<td>EU local thread checking is enabled.</td>
<td>CHV, BSW</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>14:13</th>
<th>SFR mode</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td>Access: r/w</td>
</tr>
<tr>
<td></td>
<td>Format: U2</td>
</tr>
<tr>
<td></td>
<td>This field must be zero when not in GT4(SFR) configuration i.e GTB_rendermode fuse set to SFR.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>12</th>
<th>Reserved</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td>Access: r/w</td>
</tr>
<tr>
<td></td>
<td>Format: PBC</td>
</tr>
</tbody>
</table>

| 11    | 16X16 Cross Slice Hash Disable for SF |
## GT_MODE - GT Mode Register

<table>
<thead>
<tr>
<th>Project:</th>
<th>CHV, BSW</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access:</td>
<td>r/w</td>
</tr>
<tr>
<td>Format:</td>
<td>U1</td>
</tr>
</tbody>
</table>

### Description

This field allows to control pixel block hashing across slices.

Supports 16X16 pixel block hashing in the checker-board pattern irrespective of MSAA. Setting this bit disables hashing and therefore HW will send all the Pixels to both the slices.

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>Enable [Default]</td>
<td>16X16 Checkerboard hashing enabled across slices</td>
</tr>
<tr>
<td>1h</td>
<td>Disable</td>
<td>16X16 Checkerboard hashing disabled across slices</td>
</tr>
</tbody>
</table>

### Programming Notes

Normal mode of operation in GT3 mode will be to use either 16x16 Hashing or 32x32 Hashing.

#### 10 16x16 Cross Slice Hash Disable

<table>
<thead>
<tr>
<th>Project:</th>
<th>CHV, BSW</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access:</td>
<td>r/w</td>
</tr>
<tr>
<td>Format:</td>
<td>U1</td>
</tr>
</tbody>
</table>

This field allows to control pixel block hashing across slices.

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>Enable [Default]</td>
<td>16X16 Checkerboard hashing enabled across slices</td>
</tr>
<tr>
<td>1h</td>
<td>Disable</td>
<td>16X16 Checkerboard hashing disabled across slices</td>
</tr>
</tbody>
</table>

#### 9 WIZ Hashing Mode High Bit

<table>
<thead>
<tr>
<th>Project:</th>
<th>CHV, BSW</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access:</td>
<td>r/w</td>
</tr>
<tr>
<td>Format:</td>
<td>U1</td>
</tr>
</tbody>
</table>

This field adds additional hashing modes in combination with the WIZ Hashing Mode field. The Value column in the table below refers to this field (high bit) and the WIZ Hashing Mode field (low bit). This field is don’t care if the Hashing Disable bit is set.

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>[Default]</td>
<td>8x8 Checkerboard hashing</td>
</tr>
<tr>
<td>1h</td>
<td></td>
<td>8x4 Checkerboard hashing</td>
</tr>
<tr>
<td>2h</td>
<td></td>
<td>16x4 Checkerboard hashing</td>
</tr>
<tr>
<td>3h</td>
<td></td>
<td>Reserved</td>
</tr>
</tbody>
</table>

### Programming Notes

8x4 hashing preferred for when msaa enabled
### GT_MODE - GT Mode Register

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Access</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>Reserved</td>
<td>CHV, BSW</td>
<td>r/w</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PBC</td>
</tr>
<tr>
<td>7</td>
<td>WIZ Hashing Mode</td>
<td>CHV, BSW</td>
<td>r/w</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>U1</td>
</tr>
</tbody>
</table>

**Description**: This field configures the Hashing mode in Windower. This field is don't care if the Hashing Disable bit is set. The WIZ Hashing Mode High Bit field is combined with this field to enable additional modes.

**Project**: CHV, BSW

<table>
<thead>
<tr>
<th>Field</th>
<th>Access</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>6:3</td>
<td>r/w</td>
<td>PBC</td>
</tr>
<tr>
<td>2</td>
<td>r/w</td>
<td>PBC</td>
</tr>
<tr>
<td>1:0</td>
<td>r/w</td>
<td>PBC</td>
</tr>
</tbody>
</table>
## GTSCRATCH1

### GTSCRATCH1 - GTSCRATCH1

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>4F100h</td>
</tr>
</tbody>
</table>

These (8 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these register is defined by software architecture.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>GT_Scratchpad</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these register is defined by software architecture.
### GTSCRATCH2 - GTSCRATCH2

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>4F104h</td>
</tr>
</tbody>
</table>

These (8 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these register is defined by software architecture.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>GT_Scratchpad</strong></td>
</tr>
</tbody>
</table>

- **Default Value:** 00000000h
- **Access:** R/W

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these register is defined by software architecture.
GTSCRATCH3

<table>
<thead>
<tr>
<th>GTSCRATCH3 - GTSCRATCH3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 4F108h</td>
</tr>
</tbody>
</table>

These (8 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these register is defined by software architecture.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>GT_Scratchpad</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these register is defined by software architecture.
**GTSCRATCH4**

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>GT_Scratchpad</strong></td>
</tr>
</tbody>
</table>

- Default Value: 00000000h
- Access: R/W

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these registers is defined by software architecture.
# GTSCRATCH5

<table>
<thead>
<tr>
<th><strong>GTSCRATCH5 - GTSCRATCH5</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 4F110h</td>
</tr>
</tbody>
</table>

These (8 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these register is defined by software architecture.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>GT_Scratchpad</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these register is defined by software architecture.
## GTSCRATCH6 - GTSCRATCH6

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>GT_Scratchpad</strong></td>
</tr>
</tbody>
</table>

- **Default Value:** 00000000h
- **Access:** R/W

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these registers is defined by software architecture.

- Register Space: MMIO: 0/2/0
- Project: CHV, BSW
- Source: PRM
- Default Value: 0x00000000
- Size (in bits): 32
- Address: 4F114h
GTSCRATCH7

<table>
<thead>
<tr>
<th><strong>GTSCRATCH7 - GTSCRATCH7</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 4F118h</td>
</tr>
</tbody>
</table>

These (8 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these register is defined by software architecture.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>GT_Scratchpad</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these register is defined by software architecture.
GTSCRATCH8

<table>
<thead>
<tr>
<th>GTSCRATCH8 - GTSCRATCH8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 4F11Ch</td>
</tr>
</tbody>
</table>

These (8 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these registers is defined by software architecture.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>GT_Scratchpad</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these registers is defined by software architecture.
**GTSCRATCHPAD0**

<table>
<thead>
<tr>
<th>GTSCRATCHPAD0 - GTSCRATCHPAD0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 130040h</td>
</tr>
</tbody>
</table>

GT scratchpad registers.
Scratchpad register can be R/W by both driver and GT.
One usage is that GT updates those registers before going to RC6 and reads them on RC6 exit.
Write-able and read-able from GT.
Reads do NOT rely on the register dispatch path, as dependencies could result.
Read-able and write-able from IA.
New for CHV, BSW.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>GT_Scratch0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

GT scratchpad register.
## GTSCRATCHPAD1 - GTSCRATCHPAD1

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>130044h</td>
</tr>
</tbody>
</table>

GT scratchpad registers.
Scratchpad register can be R/W by both driver and GT.
One usage is that GT updates those registers before going to RC6 and reads them on RC6 exit.
Write-able and read-able from GT.
Reads do NOT rely on the register dispatch path, as dependencies could result.
Read-able and write-able from IA.
New for CHV, BSW.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>GT_Scratch1</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GT scratchpad register.</td>
</tr>
</tbody>
</table>
GTSCRATCHPAD2

Register Space: MMIO: 0/2/0
Project: CHV, BSW
Source: PRM
Default Value: 0x00000000
Size (in bits): 32
Address: 130048h

GT scratchpad registers.
Scratchpad register can be R/W by both driver and GT.
One usage is that GT updates those registers before going to RC6 and reads them on RC6 exit.
Write-able and read-able from GT.
Reads do NOT rely on the register dispatch path, as dependencies could result.
Read-able and write-able from IA.
New for CHV, BSW.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>GT_Scratch2</strong></td>
</tr>
</tbody>
</table>

Default Value: 00000000h
Access: R/W

GT scratchpad register.
GTSCRATCHPAD3

<table>
<thead>
<tr>
<th>GTSCRATCHPAD3 - GTSCRATCHPAD3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 13004Ch</td>
</tr>
</tbody>
</table>

GT scratchpad registers.
Scratchpad register can be R/W by both driver and GT.
One usage is that GT updates those registers before going to RC6 and reads them on RC6 exit.
Write-able and read-able from GT.
Reads do NOT rely on the register dispatch path, as dependencies could result.
Read-able and write-able from IA.
New for CHV, BSW.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>GT_Scratch3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

GT scratchpad register.
**GTSCRATCHPAD4**

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>GT_Scratch4</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GT scratchpad register.</td>
</tr>
</tbody>
</table>

GT scratchpad registers. Scratchpad register can be R/W by both driver and GT. One usage is that GT updates those registers before going to RC6 and reads them on RC6 exit. Write-able and read-able from GT. Reads do NOT rely on the register dispatch path, as dependencies could result. Read-able and write-able from IA. New for CHV, BSW.
## GTSCRATCHPAD5

<table>
<thead>
<tr>
<th>GTSCRATCHPAD5 - GTSCRATCHPAD5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space:</td>
</tr>
<tr>
<td>Project:</td>
</tr>
<tr>
<td>Source:</td>
</tr>
<tr>
<td>Default Value:</td>
</tr>
<tr>
<td>Size (in bits):</td>
</tr>
<tr>
<td>Address:</td>
</tr>
</tbody>
</table>

GT scratchpad registers.
Scratchpad register can be R/W by both driver and GT.
One usage is that GT updates those registers before going to RC6 and reads them on RC6 exit.
Write-able and read-able from GT.
Reads do NOT rely on the register dispatch path, as dependencies could result.
Read-able and write-able from IA.
New for CHV, BSW.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>GT_Scratch5</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Default Value:</th>
<th>00000000h</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
</tbody>
</table>

GT scratchpad register.
GTSCRATCHPAD6

GTSCRATCHPAD6 - GTSCRATCHPAD6

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>130058h</td>
</tr>
</tbody>
</table>

GT scratchpad registers.
Scratchpad register can be R/W by both driver and GT.
One usage is that GT updates those registers before going to RC6 and reads them on RC6 exit.
Write-able and read-able from GT.
Reads do NOT rely on the register dispatch path, as dependencies could result.
Read-able and write-able from IA.
New for CHV, BSW.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>GT_Scratch6</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

GT scratchpad register.
**GTSCRATCHPAD7**

<table>
<thead>
<tr>
<th>GTSCRATCHPAD7 - GTSCRATCHPAD7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project:                      CHV, BSW</td>
</tr>
<tr>
<td>Source:                       PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 13005Ch</td>
</tr>
</tbody>
</table>

GT scratchpad registers.
Scratchpad register can be R/W by both driver and GT.
One usage is that GT updates those registers before going to RC6 and reads them on RC6 exit.
Write-able and read-able from GT.
Reads do NOT rely on the register dispatch path, as dependencies could result.
Read-able and write-able from IA.
New for CHV, BSW.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>GT_Scratch7</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

GT scratchpad register.
# GTT Cache Enable

<table>
<thead>
<tr>
<th>GTT CACHE_EN - GTT Cache Enable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000 CHV, BSW</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 04024h</td>
</tr>
</tbody>
</table>

Enable GTT Cache for respective client(s), A0: Must program/observed this to all 0 due to Big Pages Bug 1898112

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>GTT Cache Enable for CS</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Enable GTT Caching for all the client(s) below:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>31: BLIT Engine (overrides individual enables of the units)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>30: VEBX Engine (overrides individual enables of the units)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>29: MFX Engine (overrides individual enables of the units)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>28: GFX Engine (overrides individual enables of the units)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>27-15: Reserved</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>VMCunit</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>VLFunit</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>BLBunit</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>VFWunit</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>VEOunit</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>HIZunit</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>RCZunit</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>RCCunit</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>ISCunit</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>DCunit</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>MTunit</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>SOLunit</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>VFunit</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>RSunit</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>CSunit</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00000000h</td>
<td>[Default]</td>
<td>CHV, BSW</td>
</tr>
</tbody>
</table>
### GTTMMADR_LSB

**Register Space:** PCI: 0/2/0  
**Project:** CHV, BSW  
**Source:** PRM  
**Default Value:** 0x00000004  
**Size (in bits):** 32

**Address:** 00010h

Gfx Memory Mapped Address Range. This is the base address for all memory mapped registers and GTT table. This register requests allocation for the combined Graphics Translation Table Modification Range and Memory Mapped Range. For the Global GTT, this range is defined as a memory BAR in graphics device config space. It is an alias into which software is required to write Page Table Entry values (PTEs). Software may read PTE values from the global Graphics Translation Table (GTT). PTEs cannot be written directly into the global GTT memory area.

GTTMMADR comprises of 2 regions, GTT and MMADR. Prior to this feature it had a max size of 4M with 2M MMADR + {0,1M, 2M} GTT. Now GTTMMADR will have a max size of 16M. GTTMMADR = 2M MMADR + 6M rsvd + {0, 2M, 4M, 8M} GTT. GTTMMADR will be 16M aligned.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>MBA_LSB</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Memory Base Address (MBA): Set by the OS, these bits correspond to address signals [35:24]. 16MB combined for MMIO and Global GTT table aperture (2MB for MMIO, 6MB reserved and 8 MB for GTT).</td>
</tr>
<tr>
<td>23:4</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RSVD:Hardwired to 0 to indicate at least 4MB address range.</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Prefetchable Memory (PREFMEM): Hardwired to 0 to prevent prefetching.</td>
</tr>
<tr>
<td>2:1</td>
<td></td>
<td><strong>MEMTYP</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 10b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Memory Type (MEMTYP):</td>
</tr>
<tr>
<td>GTTMMADR_LSB - GTTMMADR_LSB</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>----------------------------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>00:</td>
<td>To indicate 32 bit base address</td>
<td></td>
</tr>
<tr>
<td>01: Reserved</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10:</td>
<td>To indicate 64 bit base address</td>
<td></td>
</tr>
<tr>
<td>11: Reserved</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>Reserved</td>
<td></td>
</tr>
</tbody>
</table>
### GTTMMADR_MSB

<table>
<thead>
<tr>
<th>Description</th>
<th>DWord</th>
<th>Bit</th>
<th>Access</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>31:4</td>
<td>R/W</td>
<td>00000000h</td>
<td>MBA_MSB28</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>This field must be set to 0 since addressing above 64GB is not supported.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Made them spare RW bits.</td>
</tr>
<tr>
<td></td>
<td>3:0</td>
<td>MBA_MSB</td>
<td>R/W</td>
<td>0h</td>
<td>Memory Base Address (MBA): Set by the OS, these bits correspond to address signals [35:24].16MB combined for MMIO and Global GTT table aperture (2MB for MMIO, 6MB reserved and 8 MB for GTT).</td>
</tr>
</tbody>
</table>

Gfx Memory Mapped Address Range. This is the base address for all memory mapped registers and GTT table.

This register requests allocation for the combined Graphics Translation Table Modification Range and Memory Mapped Range. For the Global GTT, this range is defined as a memory BAR in graphics device config space. It is an alias into which software is required to write Page Table Entry values (PTEs). Software may read PTE values from the global Graphics Translation Table (GTT). PTEs cannot be written directly into the global GTT memory area.

GTTMMADR comprises of 2 regions, GTT and MMADR. Prior to this feature it had a max size of 4M with 2M MMADR + {0,1M, 2M} GTT. Now GTTMMADR will have a max size of 16M. GTTMMADR = 2M MMADR + 6M rsvd + {0, 2M, 4M, 8M} GTT. GTTMMADR will be 16M aligned.
# GU_CTL0

## GU_CTL0 - GU_CTL0

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31</td>
<td><strong>GT_IOSFSB_P_READ_POLICY</strong></td>
<td>0b</td>
<td>R/W</td>
</tr>
</tbody>
</table>
|       |     | During an SOIx flow, the Punit must read Gunit registers. 0 (Default) : Allow IOSF SB P reads. Note : Punit knows when appropriate times are to issue IOSF SB P reads. 1 : Abort IOSF SB P reads.  
Note : This default is the opposite of the IOSF SB polarity bit 12_0008h[31].  
Note : For CHV, BSW, this policy register will applies to new PM endpoint (which includes the new PM endpoint). |
| 30:16 | SB Doorbellprevention |
|       | Default Value:              | 0000h         | R/W    |
|       | Access:                     |               |        |
|       | Spare register bits for future use |
| 15    | SPARE1 | | 0b | R/W |
|       | | 0 (default) : IOSF P to IOSF SB doorbell registers can be accesses by IOSF SB and IOSF SBp. Note : This should never occur during normal operation. 1 : IOSF SB and IOSF SBp accesses targeting the IOSF P to IOSF SB doorbell registers are aborted. |
| 14    | SPARE1 | | 0b | R/W |
|       | Spare register bits for future use |
| 13    | Reserved | | | |

Register Space: MMIO: 0/2/0  
Project: CHV, BSW  
Source: PRM  
Default Value: 0x00000070  
Size (in bits): 32  
Address: 182030h  
Gunit Configuration bits.
## GU_CTL0 - GU_CTL0

<table>
<thead>
<tr>
<th>Bit</th>
<th>Field</th>
<th>Default Value</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>CMDperCLK_ECO</td>
<td>0b</td>
<td>R/W</td>
<td>Default Value: 0b Access: R/W 0 (default): GSA command per clock supported. 1: A GAM write followed by a read command cannot dispatch in consecutive clocked.</td>
</tr>
<tr>
<td>11</td>
<td>Reserved</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10:9</td>
<td>SPARE2</td>
<td>00b</td>
<td>R/W</td>
<td>Default Value: 00b Access: R/W Spare bits for future use</td>
</tr>
<tr>
<td>8</td>
<td>RESERVED</td>
<td>0b</td>
<td>RO</td>
<td>Default Value: 0b Access: RO Reserved</td>
</tr>
<tr>
<td>7</td>
<td>IOSFP_DCR_Policy</td>
<td>0b</td>
<td>R/W</td>
<td>Default Value: 0b Access: R/W IOSF Primary Data Credit Policy. 0 (default) : IOSF Primary data credits reflect data buffer depth. 1: IOSF Primary infinite data credits for both posted and non-posted data is advertised. Credit accounting (and flow control) will rely on cmd credits only.</td>
</tr>
<tr>
<td>6:4</td>
<td>IOSFP_PCRD</td>
<td>111b</td>
<td>R/W</td>
<td>Default Value: 111b Access: R/W IOSF Primary Credit AdvertisementNumber of posted command (and data credits/4) will advertize.</td>
</tr>
<tr>
<td>3</td>
<td>SPARE3</td>
<td>0b</td>
<td>R/W</td>
<td>Default Value: 0b Access: R/W Spare bits for future use</td>
</tr>
<tr>
<td>2</td>
<td>Reserved</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Reserved</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>PFI_Snoop_Override</td>
<td>0b</td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td>GU_CTL0 - GU_CTL0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Access:</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 : Observe GAM snoop/non-snoop indications.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 : Force all GAM requests to snoop.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Access:</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>R/W</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Hardware Status Mask Register

<table>
<thead>
<tr>
<th>HWSTAM - Hardware Status Mask Register</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: RenderCS</td>
</tr>
<tr>
<td>Default Value: 0xFFFFFFFF</td>
</tr>
<tr>
<td>Access: R/W, RO</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Trusted Type: 1</td>
</tr>
<tr>
<td>Address: 02098h</td>
</tr>
<tr>
<td>Address: 12098h-1209Bh</td>
</tr>
<tr>
<td>Name: Hardware Status Mask Register</td>
</tr>
<tr>
<td>ShortName: HWSTAM_VCSUNIT0</td>
</tr>
<tr>
<td>Address: 1A098h-1A09Bh</td>
</tr>
<tr>
<td>Name: Hardware Status Mask Register</td>
</tr>
<tr>
<td>ShortName: HWSTAM_VECSUNIT</td>
</tr>
<tr>
<td>Address: 1C098h-1C09Bh</td>
</tr>
<tr>
<td>Name: Hardware Status Mask Register</td>
</tr>
<tr>
<td>ShortName: HWSTAM_VCSUNIT1</td>
</tr>
<tr>
<td>Address: 22098h-2209Bh</td>
</tr>
<tr>
<td>Name: Hardware Status Mask Register</td>
</tr>
<tr>
<td>ShortName: HWSTAM_BCSUNIT</td>
</tr>
</tbody>
</table>

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

**Programming Notes**

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Hardware Status Mask Register Value</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: FFFFFFFFh</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: Array of Masks</td>
</tr>
</tbody>
</table>

Refer to the Interrupt Control Register section for bit definitions. Reserved bits are RO.
# Hardware Status Page Address Register

<table>
<thead>
<tr>
<th>HWS_PGA - Hardware Status Page Address Register</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Access: R/W</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Trusted Type: 1</td>
</tr>
</tbody>
</table>

| Address: 02080h-02083h                        |
| Name: Hardware Status Page Address Register |
| ShortName: HWS_PGA_RCSUNIT                   |

| Address: 12080h-12083h                        |
| Name: Hardware Status Page Address Register |
| ShortName: HWS_PGA_VCSUNIT0                  |

| Address: 1A080h-1A083h                        |
| Name: Hardware Status Page Address Register |
| ShortName: HWS_PGA_VECSUNIT                  |

| Address: 1C080h-1C083h                        |
| Name: Hardware Status Page Address Register |
| ShortName: HWS_PGA_VCSUNIT1                  |

| Address: 22080h-22083h                        |
| Name: Hardware Status Page Address Register |
| ShortName: HWS_PGA_BCSUNIT                   |

<table>
<thead>
<tr>
<th>Description</th>
<th>Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>This register is used to program the 4 KB-aligned System Memory address of</td>
<td>CHV,</td>
</tr>
<tr>
<td>the Hardware Status Page used to report hardware status into (typically</td>
<td>BSW</td>
</tr>
<tr>
<td>cacheable) System Memory.</td>
<td></td>
</tr>
<tr>
<td>The address in this register is translated using the Global GTT in memory.</td>
<td></td>
</tr>
<tr>
<td>The mapping type of the GTT entry determines the snoop nature of the</td>
<td></td>
</tr>
<tr>
<td>transaction to memory.</td>
<td></td>
</tr>
</tbody>
</table>

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

**Programming Notes**

If the Per-Process Virtual Address Space and Exec List Enable bit is set, HW requires that the status page is programmed to allow for the context switch status to be reported.
### HCP CABAC Status

#### HCP_CABAC_STATUS - HCP CABAC Status

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:12</td>
<td>Reserved</td>
<td>MBZ</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>Temporal Direction Motion Vector Out-of-Bound Error</td>
<td></td>
<td>This flag indicates motion vectors calculated from the Temporal Direct Motion vector is larger than the allowed range.</td>
</tr>
<tr>
<td>10:7</td>
<td></td>
<td>Reserved</td>
<td>MBZ</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>Motion Vector Delta SE</td>
<td>U1</td>
<td>This flag indicates out-of-bound motion vector delta SEs coded in the bit-stream.</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>Delta QP SE</td>
<td>U1</td>
<td>This flag indicates leading-one overflow during CABAC decode of cu_qp_delta_abs.</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>Residual Error</td>
<td>RO</td>
<td></td>
</tr>
</tbody>
</table>

- **Register Space:** MMIO: 0/2/0
- **Project:** CHV, BSW
- **Source:** VideoCS
- **Default Value:** 0x00000000
- **Size (in bits):** 32
- **Trusted Type:** 1
- **Address:** 1E904h
# HCP_CABAC_STATUS - HCP CABAC Status

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Slice and Error</td>
<td>0</td>
<td>RO</td>
<td>U1</td>
</tr>
<tr>
<td></td>
<td>This flag indicates a premature end to the slice or an inconsistent end of slice on the last Ctb of a slice.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2:1</td>
<td>Reserved</td>
<td></td>
<td>MBZ</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>Ctb Concealment Flag</td>
<td>0</td>
<td>RO</td>
<td>U1</td>
</tr>
<tr>
<td></td>
<td>Each pulse from this flag indicates one Ctb is concealed by the HCP.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## HCP Decode Status

### HCP_DEC_STATUS - HCP Decode Status

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>VideoCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>RO</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Trusted Type:</td>
<td>1</td>
</tr>
<tr>
<td>Address:</td>
<td>1E900h</td>
</tr>
</tbody>
</table>

HCP Decode status.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:18</td>
<td><strong>Number of Ctbs Concealed</strong></td>
<td>0</td>
<td>U14</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This 16-bit field indicates the number of Ctbs concealed during the decoding of the current frame. This field is cleared with the HCP_PIPE_MODE_SELECT command.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td></td>
<td><strong>Frame Dec Active</strong></td>
<td>0</td>
<td>U1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This flag indicates that the decoder hardware is actively decoding a picture.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
<td><strong>Indirect Bitstream ObjectAccess Upper Bound Error</strong></td>
<td>0</td>
<td>U1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This flag indicates that the upper bound bit-stream address was reached.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15:0</td>
<td></td>
<td><strong>Bit-stream Error Flags</strong></td>
<td>0</td>
<td>U16</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This 16-bit field indicates the number of bit stream errors detected for each bit field indicated in the CABAC Status register.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
HCP Picture Checksum cIdx0

<table>
<thead>
<tr>
<th>HCP_PICTURE_CHECKSUM_CIDX0 - HCP Picture Checksum cIdx0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: VideoCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Access: RO</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Trusted Type: 1</td>
</tr>
<tr>
<td>Address: 1E91Ch</td>
</tr>
</tbody>
</table>

- The HCP Picture Checksum cIdx0 register reports the 32-bit unsigned picture checksum for cIdx=0 calculated by the HCP hardware and whose algorithm is defined in Annex D of the HEVC standard specification.
- This calculated value is updated at the end of the frame.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>Picture checksum cIdx0</td>
</tr>
</tbody>
</table>

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Value:</td>
<td>0</td>
</tr>
<tr>
<td>Format:</td>
<td>U32</td>
</tr>
</tbody>
</table>
HCP Picture Checksum cldx1

<table>
<thead>
<tr>
<th>HCP_PICTURE_CHECKSUM_CIDX1 - HCP Picture Checksum cldx1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: VideoCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Access: RO</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Trusted Type: 1</td>
</tr>
</tbody>
</table>

Address: 1E920h

- The HCP Picture Checksum cldx1 register reports the 32-bit unsigned picture checksum for cldx=1 calculated by the HCP hardware and whose algorithm is defined in Annex D of the HEVC standard specification.
- This calculated value is updated at the end of the frame.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>Picture checksum cldx1</td>
</tr>
</tbody>
</table>

- Default Value: 0
- Format: U32
### HCP Picture Checksum cIdx2

<table>
<thead>
<tr>
<th><strong>HCP_PICTURE_CHECKSUM_CIDX2</strong> - HCP Picture Checksum cIdx2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: VideoCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Access: RO</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Trusted Type: 1</td>
</tr>
<tr>
<td>Address: 1E924h</td>
</tr>
</tbody>
</table>

- The HCP Picture Checksum cIdx2 register reports the 32-bit unsigned picture checksum for cIdx=2 calculated by the HCP hardware and whose algorithm is defined in Annex D of the HEVC standard specification.
- This calculated value is updated at the end of the frame.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Picture checksum cIdx2</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: U32</td>
</tr>
</tbody>
</table>
# HDR - HDR

| Register Space: | PCI: 0/2/0 |
| Source: | PRM |
| Default Value: | 0x00000000 |
| Size (in bits): | 32 |
| Address: | 0000Ch |

## Header Type

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td>RESERVED</td>
</tr>
<tr>
<td></td>
<td>Default Value:</td>
<td>00h</td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td></td>
<td>MULTI_FUNCTION_STATUS</td>
</tr>
<tr>
<td></td>
<td>Default Value:</td>
<td>0b</td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td>MFUNC: Integrated graphics is a single function</td>
<td></td>
</tr>
<tr>
<td>22:16</td>
<td></td>
<td>HEADER_CODE</td>
</tr>
<tr>
<td></td>
<td>Default Value:</td>
<td>00h</td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td>HDR: Indicates a type 0 configuration space header format</td>
<td></td>
</tr>
<tr>
<td>15:0</td>
<td></td>
<td>RESERVED</td>
</tr>
<tr>
<td></td>
<td>Default Value:</td>
<td>0000h</td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td>Reserved</td>
<td></td>
</tr>
</tbody>
</table>
HEVC Local APIC Retry Vector

<table>
<thead>
<tr>
<th>HEVC_LAPIC_RETRY_VECT - HEVC Local APIC Retry Vector</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: VideoCS</td>
</tr>
<tr>
<td>Default Value: 0x0000000000</td>
</tr>
<tr>
<td>Access: RO</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 0D594h</td>
</tr>
<tr>
<td>Valid Projects: CHV, BSW</td>
</tr>
</tbody>
</table>

Holds the 4 last retry interrupt vectors. Retries are errors in hardware and are not expected. No interrupt is actually retried, and the interrupt stimulus will be lost if a retry occurs. The system will hang eventually. A 2-bit counter (starting at reset value of 0) is used to point to the slot/byte location from which to load the next retry vector (into the 4 available slots) in sequence. This means if a 5th retry vector shows up, it will be loaded into slot 0 again (as the counter wraps around), over-writing the retry vector which existed there in slot 0.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td>Vector Slot 3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: U8</td>
</tr>
<tr>
<td>0</td>
<td>23:16</td>
<td>Vector Slot 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: U8</td>
</tr>
<tr>
<td>0</td>
<td>15:8</td>
<td>Vector Slot 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: U8</td>
</tr>
<tr>
<td>0</td>
<td>7:0</td>
<td>Vector Slot 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: U8</td>
</tr>
</tbody>
</table>
HS Invocation Counter

**HS_INVOCATION_COUNT - HS Invocation Counter**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>RenderCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000, 0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>64</td>
</tr>
<tr>
<td>Trusted Type:</td>
<td>1</td>
</tr>
</tbody>
</table>

Address: 02300h

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
</table>
| 0     | 63:32 | **HS Invocation Count UDW** 
Number of patch objects processed by the HS stage. Updated only when HS Enable and HS Statistics Enable are set in 3DSTATE_HS |
|       | 31:0 | **HS Invocation Count LDW** 
Number of patch objects processed by the HS stage. Updated only when HS Enable and HS Statistics Enable are set in 3DSTATE_HS |
# IA32 MTRR FIX4K_C0000 High

<table>
<thead>
<tr>
<th>MTRR_FIX4K_C0000_H - IA32 MTRR FIX4K_C0000 High</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 0F13Ch</td>
</tr>
</tbody>
</table>

Fixed MTRR to identify (C0000-C8000h).

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Range0 to Range7 Memory Type</strong></td>
<td>00000000h</td>
<td>R/W</td>
</tr>
</tbody>
</table>

- Bit[63:56]: Identifies the memory type 00h-FFh of range#7.
- Bit[55:48]: Identifies the memory type 00h-FFh of range#6.
- Bit[47:40]: Identifies the memory type 00h-FFh of range#5.
- Bit[39:32]: Identifies the memory type 00h-FFh of range#4.
## IA32 MTRR FIX4K_C0000 Low

### MTRR_FIX4K_C0000_L - IA32 MTRR FIX4K_C0000 Low

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0F138h</td>
</tr>
</tbody>
</table>

Fixed MTRR to identify (C0000-C8000h).

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Range0 to Range7 Memory Type</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[31:24]: Identifies the memory type 00h-FFh of range#3.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[23:16]: Identifies the memory type 00h-FFh of range#2.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[15:8]: Identifies the memory type 00h-FFh of range#1.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[7:0]: Identifies the memory type 00h-FFh of range#0.</td>
</tr>
</tbody>
</table>
### IA32 MTRR FIX4K_C8000 High

<table>
<thead>
<tr>
<th><strong>MTRR_FIX4K_C8000_H - IA32 MTRR FIX4K_C8000 High</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 0F144h</td>
</tr>
</tbody>
</table>

Fixed MTRR to identify (C8000-D0000h).

<table>
<thead>
<tr>
<th><strong>DWord</strong></th>
<th><strong>Bit</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Range0 to Range7 Memory Type</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
</tbody>
</table>

Bit[63:56]: Identifies the memory type 00h-FFh of range#7.
Bit[55:48]: Identifies the memory type 00h-FFh of range#6.
Bit[47:40]: Identifies the memory type 00h-FFh of range#5.
Bit[39:32]: Identifies the memory type 00h-FFh of range#4.
**IA32 MTRR FIX4K_C8000 Low**

<table>
<thead>
<tr>
<th>MTRR_FIX4K_C8000_L - IA32 MTRR FIX4K_C8000 Low</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 0F140h</td>
</tr>
</tbody>
</table>

Fixed MTRR to identify (C8000-D0000h).

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Range0 to Range7 Memory Type</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td>31:24</td>
<td>Bit[31:24]: Identifies the memory type 00h-FFh of range#3.</td>
</tr>
<tr>
<td></td>
<td>23:16</td>
<td>Bit[23:16]: Identifies the memory type 00h-FFh of range#2.</td>
</tr>
<tr>
<td></td>
<td>15:8</td>
<td>Bit[15:8]: Identifies the memory type 00h-FFh of range#1.</td>
</tr>
<tr>
<td></td>
<td>7:0</td>
<td>Bit[7:0]: Identifies the memory type 00h-FFh of range#0.</td>
</tr>
</tbody>
</table>
## IA32 MTRR FIX4K_D0000 High

<table>
<thead>
<tr>
<th>MTRR_FIX4K_D0000_H - IA32 MTRR FIX4K_D0000 High</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 0F14Ch</td>
</tr>
</tbody>
</table>

Fixed MTRR to identify (D0000-D8000h)

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Range0 to Range7 Memory Type</strong></td>
</tr>
</tbody>
</table>

| Default Value: | 00000000h |
| Access:        | R/W       |

- Bit[63:56]: Identifies the memory type 00h-FFh of range#7.
- Bit[55:48]: Identifies the memory type 00h-FFh of range#6.
- Bit[47:40]: Identifies the memory type 00h-FFh of range#5.
- Bit[39:32]: Identifies the memory type 00h-FFh of range#4.
### IA32 MTRR FIX4K_D0000 Low

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Range0 to Range7 Memory Type</strong></td>
<td>00000000h</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[31:24]: Identifies the memory type 00h-FFh of range#3.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[23:16]: Identifies the memory type 00h-FFh of range#2.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[15:8]: Identifies the memory type 00h-FFh of range#1.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[7:0]: Identifies the memory type 00h-FFh of range#0.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Register Space:** MMIO: 0/2/0  
**Project:** CHV, BSW  
**Source:** PRM  
**Default Value:** 0x00000000  
**Size (in bits):** 32  

**Address:** 0F148h  

Fixed MTRR to identify (D0000-D8000h)
## IA32 MTRR FIX4K_D8000 High

### MTRR_FIX4K_D8000_H - IA32 MTRR FIX4K_D8000 High

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0F154h</td>
</tr>
</tbody>
</table>

Fixed MTRR to identify (D8000-E0000h)

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Range0 to Range7 Memory Type</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[63:56]: Identifies the memory type 00h-FFh of range#7.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[55:48]: Identifies the memory type 00h-FFh of range#6.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[47:40]: Identifies the memory type 00h-FFh of range#5.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[39:32]: Identifies the memory type 00h-FFh of range#4.</td>
</tr>
</tbody>
</table>
# IA32 MTRR FIX4K_D8000 Low

## MTRR_FIX4K_D8000_L - IA32 MTRR FIX4K_D8000 Low

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0F150h</td>
</tr>
</tbody>
</table>

Fixed MTRR to identify (D8000-E0000h)

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Range0 to Range7 Memory Type</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
</tbody>
</table>

Bit[31:24]: Identifies the memory type 00h-FFh of range#3.
Bit[23:16]: Identifies the memory type 00h-FFh of range#2.
Bit[15:8]: Identifies the memory type 00h-FFh of range#1.
Bit[7:0]: Identifies the memory type 00h-FFh of range#0.
### IA32 MTRR FIX4K_E0000 High

**MTRR_FIX4K_E0000_H - IA32 MTRR FIX4K_E0000 High**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0F15Ch</td>
</tr>
</tbody>
</table>

Fixed MTRR to identify (E0000-E8000h).

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Range0 to Range7 Memory Type</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[63:56]: Identifies the memory type 00h-FFh of range#7.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[55:48]: Identifies the memory type 00h-FFh of range#6.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[47:40]: Identifies the memory type 00h-FFh of range#5.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[39:32]: Identifies the memory type 00h-FFh of range#4.</td>
</tr>
</tbody>
</table>
## IA32 MTRR FIX4K_E0000 Low

### MTRR_FIX4K_E0000_L - IA32 MTRR FIX4K_E0000 Low

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0F158h</td>
</tr>
</tbody>
</table>

Fixed MTRR to identify (E0000-E8000h).

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Range0 to Range7 Memory Type</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[31:24]: Identifies the memory type 00h-FFh of range#3.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[23:16]: Identifies the memory type 00h-FFh of range#2.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[15:8]: Identifies the memory type 00h-FFh of range#1.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[7:0]: Identifies the memory type 00h-FFh of range#0.</td>
</tr>
</tbody>
</table>
## MTRR_FIX4K_E8000_H - IA32 MTRR FIX4K_E8000 High

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0F164h</td>
</tr>
</tbody>
</table>

Fixed MTRR to identify (E8000-F0000h).

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value:</th>
<th>Access:</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Range0 to Range7 Memory Type</strong></td>
<td>00000000h</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[63:56]: Identifies the memory type 00h-FFh of range#7.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[55:48]: Identifies the memory type 00h-FFh of range#6.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[47:40]: Identifies the memory type 00h-FFh of range#5.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[39:32]: Identifies the memory type 00h-FFh of range#4.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**IA32 MTRR FIX4K_E8000 Low**

<table>
<thead>
<tr>
<th>MTRR_FIX4K_E8000_L - IA32 MTRR FIX4K_E8000 Low</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong></td>
</tr>
<tr>
<td><strong>Project:</strong></td>
</tr>
<tr>
<td><strong>Source:</strong></td>
</tr>
<tr>
<td><strong>Default Value:</strong></td>
</tr>
<tr>
<td><strong>Size (in bits):</strong></td>
</tr>
<tr>
<td><strong>Address:</strong></td>
</tr>
</tbody>
</table>

Fixed MTRR to identify (E8000-F0000h).

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Range0 to Range7 Memory Type</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[31:24]: Identifies the memory type 00h-FFh of range#3.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[23:16]: Identifies the memory type 00h-FFh of range#2.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[15:8]: Identifies the memory type 00h-FFh of range#1.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[7:0]: Identifies the memory type 00h-FFh of range#0.</td>
</tr>
</tbody>
</table>
### IA32 MTRR FIX4K_F0000 High

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0F16Ch</td>
</tr>
</tbody>
</table>

Fixed MTRR to identify (F0000-F8000h).

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Range0 to Range7 Memory Type</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[63:56]: Identifies the memory type 00h-FFh of range#7.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[55:48]: Identifies the memory type 00h-FFh of range#6.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[47:40]: Identifies the memory type 00h-FFh of range#5.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[39:32]: Identifies the memory type 00h-FFh of range#4.</td>
</tr>
</tbody>
</table>
IA32 MTRR FIX4K_F0000 Low

<table>
<thead>
<tr>
<th>MTRR_FIX4K_F0000_L - IA32 MTRR FIX4K_F0000 Low</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 0F168h</td>
</tr>
<tr>
<td>Fixed MTRR to identify (F0000-F8000h).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Range0 to Range7 Memory Type</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

- Bit[31:24]: Identifies the memory type 00h-FFh of range#3.
- Bit[23:16]: Identifies the memory type 00h-FFh of range#2.
- Bit[15:8]: Identifies the memory type 00h-FFh of range#1.
- Bit[7:0]: Identifies the memory type 00h-FFh of range#0.
## IA32 MTRR FIX4K_F8000 High

<table>
<thead>
<tr>
<th>MTRR_FIX4K_F8000_H - IA32 MTRR FIX4K_F8000 High</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 0F174h</td>
</tr>
</tbody>
</table>

Fixed MTRR to identify (F8000-100000h).

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Range0 to Range7 Memory Type</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[63:56]: Identifies the memory type 00h-FFh of range#7.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[55:48]: Identifies the memory type 00h-FFh of range#6.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[47:40]: Identifies the memory type 00h-FFh of range#5.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[39:32]: Identifies the memory type 00h-FFh of range#4.</td>
</tr>
</tbody>
</table>
## MTRR_FIX4K_F8000_L - IA32 MTRR FIX4K_F8000 Low

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Range0 to Range7 Memory Type</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[31:24]: Identifies the memory type 00h-FFh of range#3.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[23:16]: Identifies the memory type 00h-FFh of range#2.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[15:8]: Identifies the memory type 00h-FFh of range#1.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[7:0]: Identifies the memory type 00h-FFh of range#0.</td>
</tr>
</tbody>
</table>

Register Space: MMIO: 0/2/0
Project: CHV, BSW
Source: PRM
Default Value: 0x00000000
Size (in bits): 32
Address: 0F170h

Fixed MTRR to identify (F8000-100000h).
### IA32 MTRR FIX16K_80000 High

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Range0 to Range7 Memory Type</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td>63:56</td>
<td>Bit[63:56]: Identifies the memory type 00h-FFh of range#7.</td>
</tr>
<tr>
<td></td>
<td>55:48</td>
<td>Bit[55:48]: Identifies the memory type 00h-FFh of range#6.</td>
</tr>
<tr>
<td></td>
<td>47:40</td>
<td>Bit[47:40]: Identifies the memory type 00h-FFh of range#5.</td>
</tr>
<tr>
<td></td>
<td>39:32</td>
<td>Bit[39:32]: Identifies the memory type 00h-FFh of range#4.</td>
</tr>
</tbody>
</table>
### IA32 MTRR FIX16K_80000 Low

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0F128h</td>
</tr>
</tbody>
</table>

Fixed MTRR to identify 512K-768K of the main memory (80000-A0000h).

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Range0 to Range7 Memory Type</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

- Bit[31:24]: Identifies the memory type 00h-FFh of range#3.
- Bit[23:16]: Identifies the memory type 00h-FFh of range#2.
- Bit[15:8]: Identifies the memory type 00h-FFh of range#1.
- Bit[7:0]: Identifies the memory type 00h-FFh of range#0.
## IA32 MTRR FIX16K_A0000 High

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0F134h</td>
</tr>
</tbody>
</table>

Fixed MTRR to identify 768K-1024K of the main memory (A0000-C0000h).

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Range0 to Range7 Memory Type</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

- Bit[63:56]: Identifies the memory type 00h-FFh of range#7.
- Bit[55:48]: Identifies the memory type 00h-FFh of range#6.
- Bit[47:40]: Identifies the memory type 00h-FFh of range#5.
- Bit[39:32]: Identifies the memory type 00h-FFh of range#4.
## IA32 MTRR FIX16K_A0000 Low

<table>
<thead>
<tr>
<th>MTRR_FIX16K_A0000_L - IA32 MTRR FIX16K_A0000 Low</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 0F130h</td>
</tr>
</tbody>
</table>

Fixed MTRR to identify 768K-1024K of the main memory (A0000-C0000h).

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Range0 to Range7 Memory Type</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[31:24]: Identifies the memory type 00h-FFh of range#3.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[23:16]: Identifies the memory type 00h-FFh of range#2.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[15:8]: Identifies the memory type 00h-FFh of range#1.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[7:0]: Identifies the memory type 00h-FFh of range#0.</td>
</tr>
</tbody>
</table>
### IA32 MTRR FIX64K_00000 High

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0F124h</td>
</tr>
</tbody>
</table>

Fixed MTRR to identify 0-512K of the main memory (0-80000h).

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Range0 to Range7 Memory Type</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[63:56]: Identifies the memory type 00h-FFh of range#7.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[55:48]: Identifies the memory type 00h-FFh of range#6.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[47:40]: Identifies the memory type 00h-FFh of range#5.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[39:32]: Identifies the memory type 00h-FFh of range#4.</td>
</tr>
</tbody>
</table>
## IA32 MTRR FIX64K_00000 Low

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0F120h</td>
</tr>
</tbody>
</table>

Fixed MTRR to identify 0-512K of the main memory (0-80000h).

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Range0 to Range7 Memory Type</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td>31:24</td>
<td>Identifies the memory type 00h-FFh of range#3.</td>
</tr>
<tr>
<td></td>
<td>23:16</td>
<td>Identifies the memory type 00h-FFh of range#2.</td>
</tr>
<tr>
<td></td>
<td>15:8</td>
<td>Identifies the memory type 00h-FFh of range#1.</td>
</tr>
<tr>
<td></td>
<td>7:0</td>
<td>Identifies the memory type 00h-FFh of range#0.</td>
</tr>
</tbody>
</table>
# IA32 MTRR PHYSBASE0 High

<table>
<thead>
<tr>
<th>MTRR_PHYSBASE0_H - IA32 MTRR PHYSBASE0 High</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 0F184h</td>
</tr>
</tbody>
</table>

## Variable MTRR0

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:7</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0000000000000000000000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>6:0</td>
<td>6:0</td>
<td><strong>PhysBase</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

Physical Base address[38:32] of the variable MTRR.
### MTRR_PHYSBASE0_L - IA32 MTRR PHYSBASE0 Low

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:12</td>
<td>PhysBase</td>
<td>00000h</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Physical Base address[31:0] of the variable MTRR.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11:8</td>
<td></td>
<td>Reserved</td>
<td>0000b</td>
<td>RO</td>
</tr>
<tr>
<td>7:0</td>
<td></td>
<td>Memory Type</td>
<td>00h</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Identifies the memory type 00h-FFh.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### IA32 MTRR PHYSBASE1 High

<table>
<thead>
<tr>
<th><strong>MTRR_PHYSBASE1_H</strong> - IA32 MTRR PHYSBASE1 High</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong></td>
</tr>
<tr>
<td><strong>Project:</strong></td>
</tr>
<tr>
<td><strong>Source:</strong></td>
</tr>
<tr>
<td><strong>Default Value:</strong></td>
</tr>
<tr>
<td><strong>Size (in bits):</strong></td>
</tr>
<tr>
<td><strong>Address:</strong></td>
</tr>
</tbody>
</table>

#### Variable MTRR1

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:7</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000000000000000000000000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>6:0</td>
<td>38:32</td>
<td><strong>PhysBase</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

Physical Base address[38:32] of the variable MTRR.
## IA32 MTRR PHYSBASE1 Low

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0F190h</td>
</tr>
</tbody>
</table>

### Variable MTRR1

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:12</td>
<td>PhysBase</td>
<td>00000h</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Physical Base address[31:0] of the variable MTRR.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11:8</td>
<td></td>
<td>Reserved</td>
<td>0000b</td>
<td>RO</td>
</tr>
<tr>
<td>7:0</td>
<td></td>
<td>Memory Type</td>
<td>00h</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Identifies the memory type 00h-FFh.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### MTRR_PHYSBASE2_H - IA32 MTRR PHYSBASE2 High

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0F1A4h</td>
</tr>
</tbody>
</table>

#### Variable MTRR2

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:7</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000000000000000000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>6:0</td>
<td></td>
<td><strong>PhysBase</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

Physical Base address[38:32] of the variable MTRR.
## MTRR_PHYSBASE2_L - IA32 MTRR PHYSBASE2 Low

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0F1A0h</td>
</tr>
</tbody>
</table>

### Variable MTRR2

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:12</td>
<td>PhysBase</td>
<td>00000h</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Physical Base address[31:0] of the variable MTRR.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>11:8</td>
<td>Reserved</td>
<td>0000b</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>7:0</td>
<td>Memory Type</td>
<td>00h</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Identifies the memory type 00h-FFh.
## MTRR_PHYSBASE3_H - IA32 MTRR PHYSBASE3 High

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0F1B4h</td>
</tr>
</tbody>
</table>

### Variable MTRR3

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:7</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0000000000000000000000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RO</td>
</tr>
<tr>
<td>6:0</td>
<td></td>
<td><strong>PhysBase</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>R/W</td>
</tr>
</tbody>
</table>

Physical Base address[38:32] of the variable MTRR.
### IA32 MTRR PHYSBASE3 Low

<table>
<thead>
<tr>
<th>MTRR_PHYSBASE3_L - IA32 MTRR PHYSBASE3 Low</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 0F1B0h</td>
</tr>
</tbody>
</table>

#### Variable MTRR3

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:12</td>
<td><strong>PhyBase</strong></td>
<td>00000h</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Physical Base address[31:0] of the variable MTRR.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11:8</td>
<td></td>
<td><strong>Reserved</strong></td>
<td>0000b</td>
<td>RO</td>
</tr>
<tr>
<td>7:0</td>
<td></td>
<td><strong>Memory Type</strong></td>
<td>00h</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Identifies the memory type 00h-FFh.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
IA32 MTRR PHYSBASE4 High

<table>
<thead>
<tr>
<th>MTRR_PHYSBASE4_H - IA32 MTRR PHYSBASE4 High</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 0F1C4h</td>
</tr>
</tbody>
</table>

Variable MTRR4

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:7</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000000000000000000000000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>6:0</td>
<td>PhysBase</td>
<td>Physical Base address[38:32] of the variable MTRR.</td>
</tr>
<tr>
<td></td>
<td>Default Value: 0000000b</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
<td></td>
</tr>
</tbody>
</table>

Default Value: 0000000b
Access: RO

Physical Base address[38:32] of the variable MTRR.
## MTRR_PHYSBASE4_L - IA32 MTRR PHYSBASE4 Low

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:12</td>
<td><strong>PhysBase</strong></td>
<td>00000h</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Physical Base address[31:0] of the variable MTRR.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11:8</td>
<td></td>
<td><strong>Reserved</strong></td>
<td>0000b</td>
<td>RO</td>
</tr>
<tr>
<td>7:0</td>
<td></td>
<td><strong>Memory Type</strong></td>
<td>00h</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Identifies the memory type 00h-FFh</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Register Space:** MMIO: 0/2/0  
**Project:** CHV, BSW  
**Source:** PRM  
**Default Value:** 0x00000000  
**Size (in bits):** 32  
**Address:** 0F1C0h  

Variable MTRR4
## MTRR_PHYSBASE5_H - IA32 MTRR PHYSBASE5 High

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0F1D4h</td>
</tr>
</tbody>
</table>

### Variable MTRR5

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:7</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0000000000000000000000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>6:0</td>
<td></td>
<td><strong>PhysBase</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

Physical Base address[38:32] of the variable MTRR.
# IA32 MTRR PHYSBASE5 Low

## MTRR_PHYSBASE5_L - IA32 MTRR PHYSBASE5 Low

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0F1D0h</td>
</tr>
</tbody>
</table>

Variable MTRR5

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:12</td>
<td>PhysBase</td>
<td>00000h</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Physical Base address[31:0] of the variable MTRR.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11:8</td>
<td></td>
<td>Reserved</td>
<td>0000b</td>
<td>RO</td>
</tr>
<tr>
<td>7:0</td>
<td></td>
<td>Memory Type</td>
<td>00h</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Identifies the memory type 00h-FFh.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

**PhysBase**

- Default Value: 00000h
- Access: R/W

**Reserved**

- Default Value: 0000b
- Access: RO

**Memory Type**

- Default Value: 00h
- Access: R/W
## IA32 MTRR PHYSBASE6 High

<table>
<thead>
<tr>
<th><strong>MTRR_PHYSBASE6_H - IA32 MTRR PHYSBASE6 High</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong></td>
</tr>
<tr>
<td><strong>Project:</strong></td>
</tr>
<tr>
<td><strong>Source:</strong></td>
</tr>
<tr>
<td><strong>Default Value:</strong></td>
</tr>
<tr>
<td><strong>Size (in bits):</strong></td>
</tr>
<tr>
<td><strong>Address:</strong></td>
</tr>
</tbody>
</table>

### Variable MTRR6

<table>
<thead>
<tr>
<th><strong>DWord</strong></th>
<th><strong>Bit</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:7</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td>Default Value:</td>
<td>00000000000000000000000000b</td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>RO</td>
</tr>
<tr>
<td>6:0</td>
<td><strong>PhysBase</strong></td>
<td>Physical Base address[38:32] of the variable MTRR.</td>
</tr>
<tr>
<td></td>
<td>Default Value:</td>
<td>0000000b</td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>R/W</td>
</tr>
</tbody>
</table>
### IA32 MTRR PHYSBASE6 Low

#### MTRR_PHYSBASE6_L - IA32 MTRR PHYSBASE6 Low

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x000000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0F1E0h</td>
</tr>
</tbody>
</table>

#### Variable MTRR6

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:12</td>
<td>PhysBase</td>
<td>00000h</td>
<td>R/W</td>
<td>Physical Base address[31:0] of the variable MTRR.</td>
</tr>
<tr>
<td></td>
<td>11:8</td>
<td>Reserved</td>
<td>0000b</td>
<td>RO</td>
<td></td>
</tr>
<tr>
<td></td>
<td>7:0</td>
<td>Memory Type</td>
<td>00h</td>
<td>R/W</td>
<td>Identifies the memory type 00h-FFh.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# IA32 MTRR PHYSBASE7 High

## MTRR_PHYSBASE7_H - IA32 MTRR PHYSBASE7 High

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0F1F4h</td>
</tr>
</tbody>
</table>

### Variable MTRR7

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:7</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000000000000000000000000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>6:0</td>
<td>6:0</td>
<td><strong>PhysBase</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

Physical Base address[38:32] of the variable MTRR.
### IA32 MTRR PHYSBASE7 Low

**MTRR_PHYSBASE7_L - IA32 MTRR PHYSBASE7 Low**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0F1F0h</td>
</tr>
</tbody>
</table>

#### Variable MTRR7

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:12</td>
<td><strong>PhysBase</strong></td>
<td>00000h</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Physical Base address[31:0] of the variable MTRR.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>11:8</td>
<td><strong>Reserved</strong></td>
<td>0000b</td>
<td>RO</td>
</tr>
<tr>
<td>7:0</td>
<td></td>
<td><strong>Memory Type</strong></td>
<td>00h</td>
<td>R/W</td>
</tr>
</tbody>
</table>

Identifies the memory type 00h-FFh.
IA32 MTRR PHYSBASE8 High

### MTRR_PHYSBASE8_H - IA32 MTRR PHYSBASE8 High

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0F204h</td>
</tr>
</tbody>
</table>

Variable MTRR8

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:7</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0000000000000000000000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>6:0</td>
<td></td>
<td>PhysBase</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

Physical Base address[38:32] of the variable MTRR.
### IA32 MTRR PHYSBASE8 Low

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0F200h</td>
</tr>
</tbody>
</table>

#### Variable MTRR8

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0:31:12</td>
<td>PhysBase</td>
</tr>
<tr>
<td></td>
<td>Default Value:</td>
<td>00000h</td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>Physical Base address[31:0] of the variable MTRR</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>11:8</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td>Default Value:</td>
<td>0000b</td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>RO</td>
</tr>
<tr>
<td>2</td>
<td>7:0</td>
<td>Memory Type</td>
</tr>
<tr>
<td></td>
<td>Default Value:</td>
<td>00h</td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>Identifies the memory type 00h-FFh.</td>
<td></td>
</tr>
</tbody>
</table>

**Register Space:** MMIO: 0/2/0  
**Project:** CHV, BSW  
**Source:** PRM  
**Default Value:** 0x00000000  
**Size (in bits):** 32  
**Address:** 0F200h  

---

**MTRR_PHYSBASE8_L - IA32 MTRR PHYSBASE8 Low**

---

**IA32 MTRR PHYSBASE8 Low**
# IA32 MTRR PHYSBASE9 High

**MTRR_PHYSBASE9_H - IA32 MTRR PHYSBASE9 High**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0F214h</td>
</tr>
</tbody>
</table>

Variable MTRR9

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:7</td>
<td><strong>Reserved</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value:</td>
<td>00000000000000000000000000000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
<td></td>
</tr>
<tr>
<td>6:0</td>
<td></td>
<td><strong>PhysBase</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value:</td>
<td>0000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
<td></td>
</tr>
</tbody>
</table>

Physical Base address[38:32] of the variable MTRR.
IA32 MTRR PHYSBASE9 Low

### MTRR_PHYSBASE9_L - IA32 MTRR PHYSBASE9 Low

<table>
<thead>
<tr>
<th>Register Space</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits)</td>
<td>32</td>
</tr>
<tr>
<td>Address</td>
<td>0F210h</td>
</tr>
</tbody>
</table>

#### Variable MTRR9

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:12</td>
<td>PhysBase</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Physical Base address[31:0] of the variable MTRR.</td>
</tr>
<tr>
<td>11:8</td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td>7:0</td>
<td></td>
<td>Memory Type</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Identifies the memory type 00h-FFh.</td>
</tr>
</tbody>
</table>
## IA32 MTRR PHYSMASK0 High

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space:</td>
<td>MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0F18Ch</td>
</tr>
</tbody>
</table>

### Variable MTRR0

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:7</td>
<td><strong>Reserved</strong></td>
<td>0000000000000000000000000000000000000000000000b</td>
<td>RO</td>
</tr>
<tr>
<td>6:0</td>
<td></td>
<td><strong>PhysMask</strong></td>
<td>0000000b</td>
<td>R/W</td>
</tr>
</tbody>
</table>

Physical MASK for the address[38:32] of the variable MTRR.
### IA32 MTRR PHYSMASK0 Low

#### MTRR_PHYSMASK0_L - IA32 MTRR PHYSMASK0 Low

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0F188h</td>
</tr>
</tbody>
</table>

#### Variable MTRR0

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:12</td>
<td><strong>PhysMask</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Physical MASK for the address[31:0] of the variable MTRR.</td>
</tr>
<tr>
<td>11</td>
<td>Valid</td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid bit showing that MTRR decode is active.</td>
</tr>
<tr>
<td>10:0</td>
<td><strong>Reserved</strong></td>
<td>Default Value: 0000000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
</tbody>
</table>
### MTRR_PHYSMASK1_H - IA32 MTRR PHYSMASK1 High

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0F19Ch</td>
</tr>
</tbody>
</table>

#### Variable MTRR1

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:7</td>
<td><strong>Reserved</strong></td>
<td>00000000000000000000000000000000b</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6:0</td>
<td></td>
<td><strong>PhysMask</strong></td>
<td>0000000b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access:</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Physical MASK for the address[38:32] of the variable MTRR.
# IA32 MTRR PHYSMASK1 Low

## MTRR_PHYSMASK1_L - IA32 MTRR PHYSMASK1 Low

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0F198h</td>
</tr>
</tbody>
</table>

### Variable MTRR1

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:12</td>
<td>PhysMask</td>
<td>00000h</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Physical MASK for the address[31:0] of the variable MTRR.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>Valid</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid bit showing that MTRR decode is active.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10:0</td>
<td></td>
<td>Reserved</td>
<td>00000000000b</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access:</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**IA32 MTRR PHYSMASK2 High**

<table>
<thead>
<tr>
<th>MTRR_PHYSMASK2_H - IA32 MTRR PHYSMASK2 High</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 0F1ACh</td>
</tr>
<tr>
<td><strong>Variable MTRR2</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:7</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 000000000000000000000000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>6:0</td>
<td></td>
<td><strong>PhysMask</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

Physical MASK for the address[38:32] of the variable MTRR.
### IA32 MTRR PHYSMASK2 Low

<table>
<thead>
<tr>
<th>MTRR_PHYSMASK2_L - IA32 MTRR PHYSMASK2 Low</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 0F1A8h</td>
</tr>
</tbody>
</table>

**Variable MTRR2**

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:12</td>
<td><strong>PhysMask</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Physical MASK for the address[31:0] of the variable MTRR.</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td><strong>Valid</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid bit showing that MTRR decode is active.</td>
</tr>
<tr>
<td>10:0</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
</tbody>
</table>
## MTRR_PHYSMASK3_H - IA32 MTRR PHYSMASK3 High

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:7</td>
<td><strong>Reserved</strong></td>
<td>0000000000000000000000000b</td>
<td>RO</td>
</tr>
<tr>
<td>6:0</td>
<td></td>
<td><strong>PhysMask</strong></td>
<td>0000000b</td>
<td>R/W</td>
</tr>
</tbody>
</table>

Physical MASK for the address[38:32] of the variable MTRR.
## IA32 MTRR PHYSMASK3 Low

### MTRR_PHYSMASK3_L - IA32 MTRR PHYSMASK3 Low

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0F1B8h</td>
</tr>
</tbody>
</table>

Variable MTRR3

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:12</td>
<td><strong>PhysMask</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Physical MASK for the address[31:0] of the variable MTRR.</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td><strong>Valid</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid bit showing that MTRR decode is active.</td>
</tr>
<tr>
<td>10:0</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0000000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
</tbody>
</table>

---

**IA32 MTRR PHYSMASK3 Low**

Command Reference: Registers
# IA32 MTRR PHYSMASK4 High

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0F1CCh</td>
</tr>
</tbody>
</table>

## Variable MTRR4

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:7</td>
<td><strong>Reserved</strong></td>
<td>000000000000000000000000000000000b</td>
<td>RO</td>
</tr>
<tr>
<td>6:0</td>
<td></td>
<td><strong>PhysMask</strong></td>
<td>0000000b</td>
<td>R/W</td>
</tr>
</tbody>
</table>

Physical MASK for the address[38:32] of the variable MTRR.
# IA32 MTRR PHYSMASK4 Low

## MTRR_PHYSMASK4_L - IA32 MTRR PHYSMASK4 Low

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0F1C8h</td>
</tr>
</tbody>
</table>

### Variable MTRR4

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:12</td>
<td><strong>PhysMask</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Physical MASK for the address[31:0] of the variable MTRR.</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td><strong>Valid</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid bit showing that MTRR decode is active.</td>
</tr>
<tr>
<td>10:0</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0000000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
</tbody>
</table>
## IA32 MTRR PHYSMASK5 High

<table>
<thead>
<tr>
<th>MTRR_PHYSMASK5_H - IA32 MTRR PHYSMASK5 High</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 0F1DCh</td>
</tr>
</tbody>
</table>

### Variable MTRR5

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value:</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:7</td>
<td>Reserved</td>
<td>00000000000000000000000000000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
<td></td>
</tr>
<tr>
<td>6:0</td>
<td></td>
<td>PhysMask</td>
<td>0000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
<td></td>
</tr>
</tbody>
</table>

Physical MASK for the address[38:32] of the variable MTRR.
### MTRR_PHYSMASK5_L - IA32 MTRR PHYSMASK5 Low

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>Project:</th>
<th>Source:</th>
<th>Default Value:</th>
<th>Size (in bits):</th>
</tr>
</thead>
<tbody>
<tr>
<td>MMIO: 0/2/0</td>
<td>CHV, BSW</td>
<td>PRM</td>
<td>0x00000000</td>
<td>32</td>
</tr>
</tbody>
</table>

**Address:** 0F1D8h

#### Variable MTRR5

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:12</td>
<td><strong>PhysMask</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Physical MASK for the address[31:0] of the variable MTRR.</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td><strong>Valid</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid bit showing that MTRR decode is active.</td>
</tr>
<tr>
<td>10:0</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
</tbody>
</table>
## IA32 MTRR PHYSMASK6 High

### MTRR_PHYSMASK6_H - IA32 MTRR PHYSMASK6 High

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0F1ECh</td>
</tr>
</tbody>
</table>

**Variable MTRR6**

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:7</td>
<td><strong>Reserved</strong></td>
<td>000000000000000000000000000000000b</td>
<td>RO</td>
</tr>
<tr>
<td>6:0</td>
<td></td>
<td><strong>PhysMask</strong></td>
<td>0000000b</td>
<td>R/W</td>
</tr>
</tbody>
</table>

Physical MASK for the address[38:32] of the variable MTRR.
# IA32 MTRR PHYSMASK6 Low

## MTRR_PHYSMASK6_L - IA32 MTRR PHYSMASK6 Low

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0F1E8h</td>
</tr>
</tbody>
</table>

### Variable MTRR6

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:12</td>
<td><strong>PhysMask</strong></td>
</tr>
<tr>
<td></td>
<td>Default Value:</td>
<td>00000h</td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>Physical MASK for the address[31:0] of the variable MTRR.</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td><strong>Valid</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default Value:</td>
<td>0b</td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>Valid bit showing that MTRR decode is active.</td>
<td></td>
</tr>
<tr>
<td>10:0</td>
<td><strong>Reserved</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default Value:</td>
<td>00000000000b</td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>RO</td>
</tr>
</tbody>
</table>
## IA32 MTRR PHYSMASK7 High

### MTRR_PHYSMASK7_H - IA32 MTRR PHYSMASK7 High

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
</tbody>
</table>

**Address:** 0F1FCh

**Variable MTRR7**

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:7</td>
<td>Reserved</td>
<td>00000000000000000000000000000000b</td>
<td>RO</td>
</tr>
<tr>
<td>6:0</td>
<td></td>
<td>PhysMask</td>
<td>0000000b</td>
<td>R/W</td>
</tr>
</tbody>
</table>

**Description:** Physical MASK for the address[38:32] of the variable MTRR.
# IA32 MTRR PHYSMASK7 Low

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0F1F8h</td>
</tr>
</tbody>
</table>

## Variable MTRR7

### DWord 0

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>31:12</td>
<td><strong>PhysMask</strong></td>
<td>00000h</td>
<td>R/W</td>
</tr>
</tbody>
</table>

Physical MASK for the address[31:0] of the variable MTRR.

### DWord 11

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Valid</strong></td>
<td>0b</td>
<td>R/W</td>
</tr>
</tbody>
</table>

Valid bit showing that MTRR decode is active.

### DWord 10:0

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Reserved</strong></td>
<td>00000000000b</td>
<td>RO</td>
</tr>
</tbody>
</table>
**IA32 MTRR PHYSMASK8 High**

<table>
<thead>
<tr>
<th><strong>MTRR_PHYSMASK8_H - IA32 MTRR PHYSMASK8 High</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 0F20Ch</td>
</tr>
</tbody>
</table>

Variable MTRR8

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:7</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000000000000000000000000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>6:0</td>
<td></td>
<td><strong>PhysMask</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

Physical MASK for the address[38:32] of the variable MTRR.
### IA32 MTRR PHYSMASK8 Low

**MTRR_PHYSMASK8_L - IA32 MTRR PHYSMASK8 Low**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0F208h</td>
</tr>
</tbody>
</table>

#### Variable MTRR8

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:12</td>
<td>PhysMask</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Physical MASK for the address[31:0] of the variable MTRR.</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>Valid</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid bit showing that MTRR decode is active.</td>
</tr>
<tr>
<td>10:0</td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0000000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
</tbody>
</table>
# IA32 MTRR PHYSMASK9 High

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0F21Ch</td>
</tr>
</tbody>
</table>

### Variable MTRR9

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value:</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:7</td>
<td>Reserved</td>
<td>0000000000000000000000000b</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td>6:0</td>
<td>PhysMask</td>
<td>0000000b</td>
<td>R/W</td>
</tr>
</tbody>
</table>

Physical MASK for the address[38:32] of the variable MTRR.
### IA32 MTRR PHYSMASK9 Low

<table>
<thead>
<tr>
<th>MTRR_PHYSMASK9_L - IA32 MTRR PHYSMASK9 Low</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 0F218h</td>
</tr>
</tbody>
</table>

#### Variable MTRR9

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:12</td>
<td><strong>PhysMask</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Physical MASK for the address[31:0] of the variable MTRR.</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td><strong>Valid</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid bit showing that MTRR decode is active.</td>
</tr>
<tr>
<td>10:0</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0000000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
</tbody>
</table>

---

**Variable MTRR9**

**Register Space**: MMIO: 0/2/0  
**Project**: CHV, BSW  
**Source**: PRM  
**Default Value**: 0x00000000  
**Size (in bits)**: 32  
**Address**: 0F218h
### IA Vertices Count

<table>
<thead>
<tr>
<th>IA_VERTICES_COUNT - IA Vertices Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: RenderCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000, 0x00000000</td>
</tr>
<tr>
<td>Access: R/W</td>
</tr>
<tr>
<td>Size (in bits): 64</td>
</tr>
<tr>
<td>Trusted Type: 1</td>
</tr>
<tr>
<td>Address: 02310h</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>63:32</td>
<td><strong>IA Vertices Count Report UDW</strong></td>
</tr>
<tr>
<td></td>
<td>31:0</td>
<td><strong>IA Vertices Count Report LDW</strong></td>
</tr>
</tbody>
</table>
|       |        | Total number of vertices fetched by the VF stage. This count is updated for every input vertex as long as Statistics Enable is set in VF_STATE (see the Vertex Fetch Chapter in the 3D Volume.)
## IDLE Messaging Register for Blitter Engine

### MSG_IDLE_BCS - IDLE Messaging Register for Blitter Engine

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>16</td>
</tr>
<tr>
<td>Address:</td>
<td>0800Ch</td>
</tr>
</tbody>
</table>

Message registers have bit-wise masking applied for writes. The register consists of 16 bits of data in [15:0], and 16 bits of corresponding masks in [31:16]. In order to set bit0, for example, the data would be 0x0001_0001. In order to clear bit0, for example, the data would be 0x0001_0000. Note that mask bit is the data bit offset + 16. Message registers are protected from non-GT writes via the Message Channel.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>15:6</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>RO</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>Flush and Block Acknowledgement</td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>Flush and Block Acknowledgement</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1'b0 : Not flushed and blocked &lt;default&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1'b1 : Unit has flushed and blocked its pipeline</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>Preparation for Reset Acknowledgement</td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>Go Acknowledgement</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1'b0 : Go=0 Ack &lt;default&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1'b1 : Go=1 Ack</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Requirement is that when Go=0 request is sent, the agent has to indicate busy before sending the Go=0 acknowledgement. It can only go idle again once Go=1 is received.</td>
<td></td>
</tr>
<tr>
<td>3:0</td>
<td></td>
<td>Idle Messaging</td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>Idle Messaging</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Bit[3]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Secondary Pipe Clock Gating</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1'b0 : Secondary pipe clock must be on &lt;default&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1'b1 : Secondary pipe clock may be gated</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Bit[2]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Primary Pipe Clock Gating</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1'b0 : Primary pipe clock must be on &lt;default&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1'b1 : Primary pipe clock may be gated</td>
<td></td>
</tr>
<tr>
<td>Bit[1]</td>
<td>C6 Allowed</td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>------------</td>
<td></td>
</tr>
<tr>
<td>1'b0</td>
<td>Do not allow GT to enter C6 &lt;default&gt;</td>
<td></td>
</tr>
<tr>
<td>1'b1</td>
<td>GT may enter C6</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bit[0]</th>
<th>Idle Indication</th>
</tr>
</thead>
<tbody>
<tr>
<td>1'b0</td>
<td>Pipe is busy &lt;default&gt;</td>
</tr>
<tr>
<td>1'b1</td>
<td>Pipe is idle</td>
</tr>
</tbody>
</table>

**See "Valid Combinations for Idle Messaging" Table**
**IDLE Messaging Register for Media0 Engine**

### MSG_IDLE_VCS0 - IDLE Messaging Register for Media0 Engine

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>16</td>
</tr>
<tr>
<td>Address:</td>
<td>08004h</td>
</tr>
</tbody>
</table>

Message registers have bit-wise masking applied for writes. The register consists of 16 bits of data in \([15:0]\), and 16 bits of corresponding masks in \([31:16]\). In order to set bit0, for example, the data would be 0x0001_0001. In order to clear bit0, for example, the data would be 0x0001_0000. Note that mask bit is the data bit offset + 16. Message registers are protected from non-GT writes via the Message Channel.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>15:6</td>
<td><strong>Reserved</strong></td>
<td>RO</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td><strong>Flush and Block Acknowledgement</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Flush and Block Acknowledgement</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1'b0 : Not flushed and blocked &lt;default&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1'b1 : Unit has flushed and blocked its pipeline</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td><strong>Preparation for Reset Acknowledgement</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Go Acknowledgement</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1'b0 : Go=0 Ack &lt;default&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1'b1 : Go=1 Ack</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Requirement is that when Go=0 request is sent, the agent has to indicate busy before sending the Go=0 acknowledgement. It can only idle again once Go=1 is received.</td>
<td></td>
</tr>
<tr>
<td>3:0</td>
<td></td>
<td><strong>Idle Messaging</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Idle Messaging</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[3]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Secondary Pipe Clock Gating</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1'b0 : Secondary pipe clock must be on &lt;default&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1'b1 : Secondary pipe clock may be gated</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[2]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Primary Pipe Clock Gating</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1'b0 : Primary pipe clock must be on &lt;default&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1'b1 : Primary pipe clock may be gated</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[1]</td>
<td></td>
</tr>
<tr>
<td>MSG_IDLE_VCS0 - IDLE Messaging Register for Media0 Engine</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------------------------------------------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C6 Allowed</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1'b0 : Do not allow GT to enter C6 &lt;default&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1'b1 : GT may enter C6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bit[0]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IDLE Indication - Media done</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1'b0 : Pipe is busy &lt;default&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1'b1 : Pipe is idle</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Refer to bits A024[10:9] for different Media turbo scenarios.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>** See &quot;Valid Combinations for Idle Messaging&quot; Table</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
IDLE Messaging Register for Media1 Engine

<table>
<thead>
<tr>
<th>MSG_IDLE_VCS1 - IDLE Messaging Register for Media1 Engine</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 16</td>
</tr>
<tr>
<td><strong>Address:</strong> 08008h</td>
</tr>
</tbody>
</table>

Message registers have bit-wise masking applied for writes. The register consists of 16 bits of data in [15:0], and 16 bits of corresponding masks in [31:16]. In order to set bit0, for example, the data would be 0x0001_0001. In order to clear bit0, for example, the data would be 0x0001_0000. Note that mask bit is the data bit offset + 16. Message registers are protected from non-GT writes via the Message Channel.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>15:6</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td>5</td>
<td>31:8</td>
<td><strong>Flush and Block Acknowledgement</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Flush and Block Acknowledgement</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1'b0 : Not flushed and blocked &lt;default&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1'b1 : Unit has flushed and blocked its pipeline</td>
</tr>
<tr>
<td>4</td>
<td>31:8</td>
<td><strong>Preparation for Reset Acknowledgement</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Go Acknowledgement</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1'b0 : Go=0 Ack &lt;default&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1'b1 : Go=1 Ack</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Requirement is that when Go=0 request is sent, the agent has to indicate busy before sending the Go=0 acknowledgement. It can only go idle again once Go=1 is received.</td>
</tr>
<tr>
<td>3:0</td>
<td></td>
<td><strong>Idle Messaging</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Idle Messaging</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1'b0 : Secondary pipe clock must be on &lt;default&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1'b1 : Secondary pipe clock may be gated</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[2] Primary Pipe Clock Gating</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1'b0 : Primary pipe clock must be on &lt;default&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1'b1 : Primary pipe clock may be gated</td>
</tr>
</tbody>
</table>
### MSG_IDLE_VCS1 - IDLE Messaging Register for Media1 Engine

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>C6 Allowed</td>
</tr>
<tr>
<td>1'b0: Do not allow GT to enter C6 &lt;default&gt;</td>
</tr>
<tr>
<td>1'b1: GT may enter C6</td>
</tr>
<tr>
<td>Bit[0]</td>
</tr>
<tr>
<td>Idle Indication</td>
</tr>
<tr>
<td>1'b0: Pipe is busy &lt;default&gt;</td>
</tr>
<tr>
<td>1'b1: Pipe is idle</td>
</tr>
<tr>
<td><strong>See “Valid Combinations for Idle Messaging” Table</strong></td>
</tr>
</tbody>
</table>
### IDLE Messaging Register for Render Engine

#### MSG_IDLE_CS - IDLE Messaging Register for Render Engine

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>08000h</td>
</tr>
</tbody>
</table>

Message registers have bit-wise masking applied for writes. The register consists of 16 bits of data in [15:0], and 16 bits of corresponding masks in [31:16]. In order to set bit0, for example, the data would be 0x0001_0001. In order to clear bit0, for example, the data would be 0x0001_0000. Note that mask bit is the data bit offset + 16. Message registers are protected from non-GT writes via the Message Channel.

**NOTE** - GPGPU_ACTIVE and MEDIA_ACTIVE are context saved and restored; the other bits are cleared out before saving context.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td><strong>Context Save Mask</strong></td>
<td>R/W</td>
<td>When context save is in progress, mask is forced to particular value to save off messages that need to be retained across an RC6 event. Currently for this register, the following fields are context saved and restored for an RC6 event: [7] GPGPU_ACTIVE, [6] MEDIA_ACTIVE</td>
</tr>
<tr>
<td>15:8</td>
<td></td>
<td><strong>Reserved</strong></td>
<td>RO</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td><strong>GPGPU Active Load Indication</strong></td>
<td>R/W</td>
<td>Active Load Indication - bits[7:6] of this register 2'b00 : Render load is being executed &lt;default&gt; 2'b01 : Media load is being executed 2'b10 : GPGPU load is being executed 2'b11 : Undefined gpmunit self-clears this bit upon sampling.</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td><strong>Media Active Load Indication</strong></td>
<td>R/W</td>
<td>Active Load Indication - bits[7:6] of this register 2'b00 : Render load is being executed &lt;default&gt; 2'b01 : Media load is being executed 2'b10 : GPGPU load is being executed</td>
</tr>
</tbody>
</table>
### MSG_IDLE_CS - IDLE Messaging Register for Render Engine

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2'b11</td>
<td>Undefined</td>
</tr>
<tr>
<td></td>
<td>gpmunit self-clears this bit upon sampling.</td>
</tr>
<tr>
<td>5</td>
<td><strong>Flush and Block Acknowledgement</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td>Flush and Block Acknowledgement</td>
</tr>
<tr>
<td></td>
<td>1'b0 : Not flushed and blocked &lt;default&gt;</td>
</tr>
<tr>
<td></td>
<td>1'b1 : Unit has flushed and blocked its pipeline</td>
</tr>
<tr>
<td>4</td>
<td><strong>Preparation for Reset Acknowledgement</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td>Go Acknowledgement</td>
</tr>
<tr>
<td></td>
<td>1'b0 : Go=0 Ack &lt;default&gt;</td>
</tr>
<tr>
<td></td>
<td>1'b1 : Go=1 Ack</td>
</tr>
<tr>
<td></td>
<td>Requirement is that when Go=0 request is sent, the agent has to indicate busy before sending the Go=0 acknowledgement. It can only go idle again once Go=1 is received. Also, Refer to the description for bits A024[10:9]</td>
</tr>
<tr>
<td>3:0</td>
<td><strong>Idle Messaging</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td>Idle Messaging</td>
</tr>
<tr>
<td></td>
<td>1'b0 : Secondary pipe clock must be on &lt;default&gt;</td>
</tr>
<tr>
<td></td>
<td>1'b1 : Secondary pipe clock may be gated</td>
</tr>
<tr>
<td></td>
<td>Bit[2] Primary Pipe Clock Gating</td>
</tr>
<tr>
<td></td>
<td>1'b0 : Primary pipe clock must be on &lt;default&gt;</td>
</tr>
<tr>
<td></td>
<td>1'b1 : Primary pipe clock may be gated</td>
</tr>
<tr>
<td></td>
<td>Bit[1] C6 Allowed</td>
</tr>
<tr>
<td></td>
<td>1'b0 : Do not allow GT to enter C6 &lt;default&gt;</td>
</tr>
<tr>
<td></td>
<td>1'b1 : GT may enter C6</td>
</tr>
<tr>
<td></td>
<td>Bit[0] Idle Indication</td>
</tr>
<tr>
<td></td>
<td>1'b0 : Pipe is busy &lt;default&gt;</td>
</tr>
<tr>
<td></td>
<td>1'b1 : Pipe is idle</td>
</tr>
<tr>
<td></td>
<td>** See &quot;Valid Combinations for Idle Messaging&quot; Table</td>
</tr>
<tr>
<td></td>
<td>Also, Refer to the description for bits A024[10:9]</td>
</tr>
</tbody>
</table>
**IDLE Messaging Register for VEBox**

<table>
<thead>
<tr>
<th>MSG_IDLE_VECS - IDLE Messaging Register for VEBox</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 16</td>
</tr>
<tr>
<td><strong>Address:</strong> 08010h</td>
</tr>
</tbody>
</table>

Message registers have bit-wise masking applied for writes. The register consists of 16 bits of data in [15:0], and 16 bits of corresponding masks in [31:16]. In order to set bit0, for example, the data would be 0x0001_0001. In order to clear bit0, for example, the data would be 0x0001_0000. Note that mask bit is the data bit offset + 16. Message registers are protected from non-GT writes via the Message Channel.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>15:6</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td><strong>Flush and Block Acknowledgement</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Flush and Block Acknowledgement</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1'b0 : Not flushed and blocked &lt;default&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1'b1 : Unit has flushed and blocked its pipeline</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td><strong>Preparation for Reset Acknowledgement</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Go Acknowledgement</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1'b0 : Go=0 Ack &lt;default&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1'b1 : Go=1 Ack</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Requirement is that when Go=0 request is sent, the agent has to indicate busy before sending the Go=0 acknowledgement. It can only idle again once Go=1 is received.</td>
</tr>
<tr>
<td>3:0</td>
<td></td>
<td><strong>Idle Messaging</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Idle Messaging</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[3]   <strong>Secondary Pipe Clock Gating</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1'b0 : Secondary pipe clock must be on &lt;default&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1'b1 : Secondary pipe clock may be gated</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[2]   <strong>Primary Pipe Clock Gating</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1'b0 : Primary pipe clock must be on &lt;default&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1'b1 : Primary pipe clock may be gated</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[1]   <strong>Primary Pipe Clock Gating</strong></td>
</tr>
<tr>
<td><strong>MSG_IDLE_VECS - IDLE Messaging Register for VEBox</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C6 Allowed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1'b0 : Do not allow GT to enter C6 &lt;default&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1'b1 : GT may enter C6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bit[0]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Idle Indication</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1'b0 : Pipe is busy &lt;default&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1'b1 : Pipe is idle</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>See “Valid Combinations forIdle Messaging” Table</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
IDLE Messaging Register for Wi-Di

<table>
<thead>
<tr>
<th>MSG_IDLE_WIN - IDLE Messaging Register for Wi-Di</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 16</td>
</tr>
<tr>
<td>Address: 08014h</td>
</tr>
</tbody>
</table>

Message registers have bit-wise masking applied for writes. The register consists of 16 bits of data in [15:0], and 16 bits of corresponding masks in [31:16]. In order to set bit0, for example, the data would be 0x0001_0001. In order to clear bit0, for example, the data would be 0x0001_0000. Note that mask bit is the data bit offset + 16. Message registers are protected from non-GT writes via the Message Channel.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>15:5</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>RO</td>
</tr>
<tr>
<td>4</td>
<td>3:0</td>
<td>Idle Messaging</td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>Idle Messaging Bit[3]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Secondary Pipe Clock Gating</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1'b0 : Secondary pipe clock must be on &lt;default&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1'b1 : Secondary pipe clock may be gated</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Bit[2]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Primary Pipe Clock Gating</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1'b0 : Primary pipe clock must be on &lt;default&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1'b1 : Primary pipe clock may be gated</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Bit[1]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>C6 Allowed</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1'b0 : Do not allow GT to enter C6 &lt;default&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1'b1 : GT may enter C6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Bit[0]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Idle Indication</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1'b0 : Pipe is busy &lt;default&gt;</td>
<td></td>
</tr>
<tr>
<td><strong>MSG_IDLE_WIN</strong> - IDLE Messaging Register for Wi-Di</td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1'b1 : Pipe is idle</td>
<td></td>
<td></td>
</tr>
<tr>
<td>** See &quot;Valid Combinations for Idle Messaging&quot; Table</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Idle Switch Delay

### IDLEDLY - Idle Switch Delay

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>RenderCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0223Ch</td>
</tr>
</tbody>
</table>

#### IDLEDLY Register Addresses

- **Address:** 1223Ch-1223Fh
- **Name:** Idle Switch Delay
- **ShortName:** IDLEDLY_VCSUNIT0

- **Address:** 1A23Ch-1A23Fh
- **Name:** Idle Switch Delay
- **ShortName:** IDLEDLY_VECSUNIT

- **Address:** 1C23Ch-1C23Fh
- **Name:** Idle Switch Delay
- **ShortName:** IDLEDLY_VCSUNIT1

- **Address:** 2223Ch-2223Fh
- **Name:** Idle Switch Delay
- **ShortName:** IDLEDLY_BCSUNIT

The IDLEDLY register contains an Idle Delay field which specifies the minimum number of microseconds allowed for command streamer to wait before a context is switched out leading to IDLE state in Execlist mode, i.e. following this context switch there is no active element available in HW to execute.

A default value of 0, means that by default, there is no restriction to wait on a context switch leading to IDLE. This register has no significance when Execlists are not enabled.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:21</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
<tr>
<td>20:0</td>
<td></td>
<td><strong>IDLE Delay</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: All</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: U21</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Minimum number of micro-seconds allowed</td>
</tr>
</tbody>
</table>
Indirect Context Pointer

**INDIRECT_CTX - Indirect Context Pointer**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Trusted Type:</td>
<td>1</td>
</tr>
</tbody>
</table>

**Address: 021C4h-021C7h**

**Name:** Indirect Context Pointer

**ShortName:** INDIRECT_CTX_RCSUNIT

**Address: 121C4h-121C7h**

**Name:** Indirect Context Pointer

**ShortName:** INDIRECT_CTX_VCSUNIT0

**Address: 1A1C4h-1A1C7h**

**Name:** Indirect Context Pointer

**ShortName:** INDIRECT_CTX_VECSUNIT

**Address: 1C1C4h-1C1C7h**

**Name:** Indirect Context Pointer

**ShortName:** INDIRECT_CTX_VCSUNIT1

**Address: 221C4h-221C7h**

**Name:** Indirect Context Pointer

**ShortName:** INDIRECT_CTX_BCSUNIT

This register is used to program the indirect address to be executed between CS and SVG engine context if enabled. This will only get executed due to regular context save/restore and not during power restore. This register is part of the execution list context and will be executed per context. Only supported if execution list is enabled. There is no preempting workloads within this context.

<table>
<thead>
<tr>
<th>Programming Notes</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>The following commands are not supported within Render CS indirect context:</td>
<td>RenderCS</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command Name</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>MI_WAIT_FOR_EVENT</td>
<td>RenderCS</td>
</tr>
<tr>
<td>MI_SEMAPHORE_SIGNAL</td>
<td></td>
</tr>
<tr>
<td>MI_ARB_CHECK</td>
<td></td>
</tr>
<tr>
<td>MI_RS_CONTROL</td>
<td></td>
</tr>
<tr>
<td>MI_REPORT_HEAD</td>
<td></td>
</tr>
</tbody>
</table>
## INDIRECT_CTX - Indirect Context Pointer

<p>| |</p>
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>MI_URB_ATOMIC_ALLOC</td>
</tr>
<tr>
<td>MI_SUSPEND_FLUSH</td>
</tr>
<tr>
<td>MI_TOPOLOGY_FILTER</td>
</tr>
<tr>
<td>MI_RS_CONTEXT</td>
</tr>
<tr>
<td>MI_SET_CONTEXT</td>
</tr>
<tr>
<td>MI_URB_CLEAR</td>
</tr>
<tr>
<td>MI_SEMAPHORE_WAIT in Memory Poll Mode is not supported.</td>
</tr>
<tr>
<td>MI_BATCH_BUFFER_START</td>
</tr>
<tr>
<td>MI_CONDITIONAL_BATCH_BUFFER_END</td>
</tr>
<tr>
<td>MEDIA_OBJECT_WALKER</td>
</tr>
<tr>
<td>GPGPU_WALKER</td>
</tr>
<tr>
<td>3DPRIMITIVE</td>
</tr>
<tr>
<td>3DSTATE_BINDING_TABLE_POINTERS_VS</td>
</tr>
<tr>
<td>3DSTATE_BINDING_TABLE_POINTERS_HS</td>
</tr>
<tr>
<td>3DSTATE_BINDING_TABLE_POINTERS_DS</td>
</tr>
<tr>
<td>3DSTATE_BINDING_TABLE_POINTERS_GS</td>
</tr>
<tr>
<td>3DSTATE_BINDING_TABLE_POINTERS_PS</td>
</tr>
<tr>
<td>3DSTATE_GATHER_CONSTANT_VS</td>
</tr>
<tr>
<td>3DSTATE_GATHER_CONSTANT_GS</td>
</tr>
<tr>
<td>3DSTATE_GATHER_CONSTANT_HS</td>
</tr>
<tr>
<td>3DSTATE_GATHER_CONSTANT_DS</td>
</tr>
<tr>
<td>3DSTATE_GATHER_CONSTANT_PS</td>
</tr>
<tr>
<td>3DSTATE_DX9_CONSTANTF_VS</td>
</tr>
<tr>
<td>3DSTATE_DX9_CONSTANTF_HS</td>
</tr>
<tr>
<td>3DSTATE_DX9_CONSTANTF_DS</td>
</tr>
<tr>
<td>3DSTATE_DX9_CONSTANTF_GS</td>
</tr>
<tr>
<td>3DSTATE_DX9_CONSTANTF_PS</td>
</tr>
<tr>
<td>3DSTATE_DX9_CONSTANTI_VS</td>
</tr>
<tr>
<td>3DSTATE_DX9_CONSTANTI_HS</td>
</tr>
<tr>
<td>3DSTATE_DX9_CONSTANTI_DS</td>
</tr>
<tr>
<td>3DSTATE_DX9_CONSTANTI_GS</td>
</tr>
<tr>
<td>3DSTATE_DX9_CONSTANTI_PS</td>
</tr>
<tr>
<td>3DSTATE_DX9_CONSTANTB_VS</td>
</tr>
<tr>
<td>3DSTATE_DX9_CONSTANTB_HS</td>
</tr>
<tr>
<td>3DSTATE_DX9_CONSTANTB_DS</td>
</tr>
</tbody>
</table>
### INDIRECT_CTX - Indirect Context Pointer

<table>
<thead>
<tr>
<th>Register Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>3DSTATE_DX9_CONSTANTB GS</td>
</tr>
<tr>
<td>3DSTATE_DX9_CONSTANTB PS</td>
</tr>
<tr>
<td>3DSTATE_DX9_LOCAL_VALID VS</td>
</tr>
<tr>
<td>3DSTATE_DX9_LOCAL_VALID DS</td>
</tr>
<tr>
<td>3DSTATE_DX9_LOCAL_VALID HS</td>
</tr>
<tr>
<td>3DSTATE_DX9_LOCAL_VALID GS</td>
</tr>
<tr>
<td>3DSTATE_DX9_LOCAL_VALID PS</td>
</tr>
<tr>
<td>3DSTATE_DX9_GENERATE_ACTIVE VS</td>
</tr>
<tr>
<td>3DSTATE_DX9_GENERATE_ACTIVE HS</td>
</tr>
<tr>
<td>3DSTATE_DX9_GENERATE_ACTIVE DS</td>
</tr>
<tr>
<td>3DSTATE_DX9_GENERATE_ACTIVE GS</td>
</tr>
<tr>
<td>3DSTATE_DX9_GENERATE_ACTIVE PS</td>
</tr>
<tr>
<td>3DSTATE_BINDING_TABLE_EDIT VS</td>
</tr>
<tr>
<td>3DSTATE_BINDING_TABLE_EDIT GS</td>
</tr>
<tr>
<td>3DSTATE_BINDING_TABLE_EDIT HS</td>
</tr>
<tr>
<td>3DSTATE_BINDING_TABLE_EDIT DS</td>
</tr>
<tr>
<td>3DSTATE_BINDING_TABLE_EDIT PS</td>
</tr>
<tr>
<td>3DSTATE_CONSTANT VS</td>
</tr>
<tr>
<td>3DSTATE_CONSTANT GS</td>
</tr>
<tr>
<td>3DSTATE_CONSTANT PS</td>
</tr>
<tr>
<td>3DSTATE_CONSTANT HS</td>
</tr>
<tr>
<td>3DSTATE_CONSTANT DS</td>
</tr>
<tr>
<td>MI_BATCH_BUFFER_END</td>
</tr>
</tbody>
</table>
## INDIRECT_CTX - Indirect Context Pointer

### Workaround

Workaround: [Render CS Only][Execlist Mode of Scheduling]: SW must ensure arbitration is switched off while context restore is in progress for any given context. This is achieved by disabling arbitration by programming MI_ARB_ON_OFF to “Arbitration Disable” in RCS_INDIRECT_CTX buffer and by enabling back the arbitration by programming MI_ARB_ON_OFF to “Arbitration Enable” as the last command prior to MI_BATCH_END in the BB_PER_CTX_PTR buffer of every context submitted. Note that RCS_INDIRECT_CTX_OFFSET could be set to default value or any other legitimate value as per the programming notes of the register definition. Arbitration disable by programming MI_ARB_ON_OFF (Arbitration Disabled) in RCS_INDIRECT_CTX buffer. Arbitration enabled by programming MI_ARB_ON_OFF (Arbitration Enabled) as the last command prior to MI_BATCH_BUFFER_END in BB_PER_CTX_PTR buffer. Additional Note: This WA need not be applied when it is guaranteed for no preemption to occur during execution of GPGPU workload. Preemption of GPGPU workload can be avoided by Bracketing the GPGPU workload with MI_ARB_ON_OFF (Arbitration Disable) and MI_ARB_ON_OFF (Arbitration Enable) command. MI_ARB_ON_OFF is a privileged command and can only be programmed in ring buffer or in privileged batch buffer (batch buffer in GGTT memory). Pending execlist submitted must not trigger preemption of the ongoing GPGPU workload due to following reasons First context of the pending execlist submitted is not the same as the ongoing GPGPU context. Force restore bit set for the submitted pending execlist.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:6</td>
<td><strong>Indirect CS Context Address</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: GraphicsAddress[31:6]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Pointer to the Context in memory to be executed as a batch.</td>
</tr>
<tr>
<td>5:0</td>
<td></td>
<td><strong>Size of Indirect CS Context</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: U6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This is the size of the Indirect Context for CS. This size supports up to 63 cache lines worth of commands where a cache line is 64B. If programmed to zero then the indirect fetch of the CS context is disabled.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Value</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>[0,63]</td>
</tr>
</tbody>
</table>
# Instruction Parser Mode Register

<table>
<thead>
<tr>
<th>INSTPM - Instruction Parser Mode Register</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> RenderCS</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x000004080 [CHV:B, CHV:C, CHV:K] 0x000006080 [CHV:A]</td>
</tr>
<tr>
<td><strong>Access:</strong> R/W, RO</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Trusted Type:</strong> 1</td>
</tr>
<tr>
<td><strong>Address:</strong> 020C0h</td>
</tr>
<tr>
<td><strong>Address:</strong> 120C0h-120C3h</td>
</tr>
<tr>
<td><strong>Name:</strong> Instruction Parser Mode Register</td>
</tr>
<tr>
<td><strong>ShortName:</strong> INSTPM_VCSUNIT0</td>
</tr>
<tr>
<td><strong>Address:</strong> 1A0C0h-1A0C3h</td>
</tr>
<tr>
<td><strong>Name:</strong> Instruction Parser Mode Register</td>
</tr>
<tr>
<td><strong>ShortName:</strong> INSTPM_VECSUNIT</td>
</tr>
<tr>
<td><strong>Address:</strong> 1C0C0h-1C0C3h</td>
</tr>
<tr>
<td><strong>Name:</strong> Instruction Parser Mode Register</td>
</tr>
<tr>
<td><strong>ShortName:</strong> INSTPM_VCSUNIT1</td>
</tr>
<tr>
<td><strong>Address:</strong> 220C0h-220C3h</td>
</tr>
<tr>
<td><strong>Name:</strong> Instruction Parser Mode Register</td>
</tr>
<tr>
<td><strong>ShortName:</strong> INSTPM_BCSUNIT</td>
</tr>
</tbody>
</table>

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

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td><strong>Mask Bits</strong></td>
</tr>
</tbody>
</table>

- **Format:** Mask[15:0]
- **Masks:** These bits serve as write enables for bits 15:0. If this register is written with any of these bits clear the corresponding bit in the field 15:0 will not be modified. Reading these bits always returns 0s.

### Reserved

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
</tbody>
</table>

- **Project:** CHV, BSW
- **Access:** RO
- **Format:** MBZ

### Replay Mode

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td></td>
<td><strong>Replay Mode</strong></td>
</tr>
</tbody>
</table>

- **Project:** CHV, BSW
- **Format:** U1
- This field controls the granularity of the replay mechanism when coming back into a previously preempted context.

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1h</td>
<td>Object Level Preemption</td>
<td>Object Level. Preemption is done on an Object Level Boundary in VF. Objects send down by VF are completely rendered. Pipeline is flushed before switching to the next context. On resubmission of the context VF starts parsing form the object where it got preempted last time.</td>
</tr>
<tr>
<td>0h</td>
<td>Drawcall Level</td>
<td>Pipeline is flushed before switching to the next context. Commands parsed are committed to completing before a context switch.</td>
</tr>
</tbody>
</table>

### Programming Notes

This bit ust be set to 0 prior to any 3DPRIMITVE using trifan, polygon, lineloop or quadstrip topology. This bit ust be set to 0 prior to any 3DPRIMITVE using linestrip_adjacency and 3dstate_GS.enable is set to 1.

### Reserved

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>14:13</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
</tbody>
</table>

- **Default Value:** 11b
- **Project:** CHV, BSW
- **Format:** Must Be One

### Reserved

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
</tbody>
</table>

- **Project:** CHV, BSW
- **Format:** Must Be One

### Reserved

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
</tbody>
</table>

- **Project:** CHV, BSW

### CLFLUSH Toggle
<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10</td>
<td><code>Reserved</code></td>
</tr>
<tr>
<td></td>
<td>9:8</td>
<td><code>Reserved</code></td>
</tr>
</tbody>
</table>
|       | 7   | **Force Sync Command Ordering**  
|       |     | Default Value: 1b  
|       |     | Project: CHV, BSW  
|       |     | Format: Enable  
|       |     | By default, driver/OS synchronization commands (MI_STORE_DATA_IMM, for instance) can execute out of order with respect to 3D state and 3D primitive commands. When set, this bit forces ordering of these commands. See section 3.2.2 for a list of these commands. |
|       | 6   | **CONSTANT_BUFFER Address Offset Disable**  
|       |     | Project: CHV, BSW  
|       |     | Format: Disable  
|       |     | When this bit is clear, the 3DSTATE_CONSTANT_* Buffers' Starting Address is used as a DynamicStateOffset. That is, it serves as an offset from the Dynamic State Base Address. Accesses will be subject to Dynamic State bounds checking. When this bit is set, the 3DSTATE_CONSTANT_* Buffers' Starting Address is used as a true GraphicsAddress (not an offset). No bounds checking will be performed during access. |
|       | 5   | `Reserved` |
|       | 4   | `Reserved` |
|       | 3   | **Media Instruction Disable**  
|       |     | Project: CHV, BSW  
|       |     | Format: U1  
<p>|       |     | This bit instructs the Renderer instruction parser to parse and error-check Media instructions, but not execute them. Format = Disable |</p>
<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Project</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
<td><strong>3D Rendering Instruction Disable</strong></td>
<td>CHV, BSW</td>
<td>U1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This bit instructs the Renderer instruction parser to parse and error-check 3D Rendering instructions, but not execute them. This bit must always be set by software if 3D State Instruction Disable is set. Setting this bit without setting 3D State Instruction Disable is allowed. Format = Disable</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td><strong>3D State Instruction Disable</strong></td>
<td>CHV, BSW</td>
<td>Disable</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td><strong>Texture Palette Load Instruction Disable</strong></td>
<td>CHV, BSW</td>
<td>U1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This bit instructs the Renderer instruction parser to parse and error-check Texture Palette Load instructions, but not execute them. Format = Disable</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Internal GAM State

<table>
<thead>
<tr>
<th>INTSTATE - Internal GAM State</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 040C0h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Reserved</strong></td>
</tr>
</tbody>
</table>
### INTERRUPT LINE

#### INTRLINE - INTERRUPT LINE

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>PCI: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000100</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>16</td>
</tr>
<tr>
<td>Address:</td>
<td>0003Ch</td>
</tr>
</tbody>
</table>

3C - Interrupt. This register is programmed by SBIOS. It is not used by the graphics/display driver. This 8-bit register is used to communicate interrupt line routing information. It is read/write and must be implemented by the device. POST software will write the routing information into this register as it initializes and configures the system.

The value in this register tells which input of the system interrupt controller(s) the device’s interrupt pin is connected to. The device itself does not use this value, rather it is used by device drivers and operating systems to determine priority and vector information.

3D - Interrupt. This register is programmed by SBIOS. It is not used by the graphics/display driver.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>15:8</td>
<td><strong>INTERRUPT_PIN</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 01h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IPIN: Value indicates which interrupt pin this device uses. 01h: INTA</td>
</tr>
<tr>
<td>7:0</td>
<td></td>
<td><strong>INTRLINE</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ILIN: BIOS written value to communicate interrupt line routing information to the device driver. Used to communicate interrupt line routing information. POST software writes the routing information into this register as it initializes and configures the system. The value in this register indicates to which input of the system interrupt controller the device’s interrupt pin is connected.</td>
</tr>
</tbody>
</table>
## Interrupt Mask Register

### IMR - Interrupt Mask Register

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>RenderCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0xFFFFFFFF</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W, RO</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
</tbody>
</table>

**Address:**

- **020A8h**
- **120A8h-120ABh**
- **1A0A8h-1A0ABh**
- **1C0A8h-1C0ABh**
- **220A8h-220ABh**

**Name:**

- Interrupt Mask Register
- Interrupt Mask Register
- Interrupt Mask Register
- Interrupt Mask Register

**ShortName:**

- IMR_VCSUNIT0
- IMR_VECSUNIT
- IMR_VCSUNIT1
- IMR_BCSUNIT

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Interrupt Mask Bits</strong></td>
</tr>
</tbody>
</table>

**Format:** InterruptMask[32] Refer to the Interrupt Control Register section for bit definitions.

This field contains a bit mask which selects which interrupt bits (from the ISR) are reported in the IIR. Reserved bits in the Interrupt Control Register are RO.

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FFFF FFFFh</td>
<td>[Default]</td>
<td></td>
</tr>
<tr>
<td>0h</td>
<td>Not Masked</td>
<td>Will be reported in the IIR</td>
</tr>
<tr>
<td>1h</td>
<td>Masked</td>
<td>Will not be reported in the IIR</td>
</tr>
</tbody>
</table>
## IOBAR - IOBAR

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>PCI: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000001</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>00020h</td>
</tr>
</tbody>
</table>

I/O Base Address. This is used only by SBIOS. This register is the base address for the MMIO_INDEX and MMIO_DATA registers.

This register provides the Base offset of the I/O registers within Device #2. Access to the 8Bs of IO space is allowed in PM state D0 when IO Enable (PCICMD bit 0) set. Access is disallowed:

* in PM states D1-D3 or
* if IO Enable is clear or
* if Device #2 is turned off or
* if Internal graphics is disabled thru the fuse or fuse override mechanisms.

Note that access to this IO BAR is independent of VGA functionality within Device #2.

If accesses to this IO bar is allowed then the GMCH claims all 8, 16 or 32 bit IO cycles from the CPU that falls within the 8B claimed.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value:</th>
<th>Access:</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td>RESERVED</td>
<td>0000h</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15:6</td>
<td>BASE_ADDRESS</td>
<td>Default Value: 0000h</td>
<td>Access: R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>BA:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Set by the OS, these bits correspond to address signals [15:6]. IOBAR is to be used for both GTLC register programming and GTT table programming. This is an indirect access method.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5:1</td>
<td>RESERVED</td>
<td>Default Value: 0h</td>
<td>Access: RO</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>RESOURCE_TYPE_RTE</td>
<td>Default Value: 1b</td>
<td>Access: RO</td>
<td></td>
</tr>
</tbody>
</table>

Indicates a request for I/O space
## L3 Bank Status

### L3STAT - L3 Bank Status

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0B128h</td>
</tr>
</tbody>
</table>

#### L3 Status register

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31</td>
<td>L3 Fill Access Status bit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This register is Hardware Set and Clear.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Set condition: set when the first command is seen on LTCC-LTCD interface.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reset condition: reset when the first Pipeline Flush command is seen on the LTCC-LTCD interface.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reset condition: This Flag will be reset only if we have atleast 1 modified line in the cache written by DC client.</td>
</tr>
<tr>
<td>30:0</td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
</tbody>
</table>

---

*Command Reference: Registers*
## L3CD Error Status register 1

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:25</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td>24</td>
<td></td>
<td><strong>Double bit ECC error detected</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W One Clear</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Indicates if bank detected a double bit ECC error. When <code>ltcd_lbcf_ecc_2bit_err_valid</code> is set.</td>
</tr>
<tr>
<td>23:14</td>
<td></td>
<td><strong>Parity row address error</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 0000000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W One Clear</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Data array address which has parity B1. Report the data array address which has the Error. <code>ltcd_lbcf_parity_err_rownum[9:0]</code>. Once set by HW, it can be cleared only by MMIO Write of 1 to this register bit 13. Driver needs to write 1 to clear this bit.</td>
</tr>
<tr>
<td>13</td>
<td></td>
<td><strong>Parity Error Valid</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W One Clear</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Parity Error valid. Report the Parity Error. <code>ltcd_lbcf_parity_err_valid</code>. Once set by HW, it can be cleared only by MMIO Write of 1 to this register bit 13. Driver needs to write 1 to clear this bit. when <code>ltcd_lbcf_parity_err_valid</code> is asserted, lbcf generates interrupt to <code>ltiseqsl_lbcf_ltiseqsl_parity_intr</code>.</td>
</tr>
<tr>
<td>12:11</td>
<td></td>
<td><strong>Parity error bank number</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W One Clear</td>
</tr>
<tr>
<td></td>
<td></td>
<td>bank number which has parity error. Report the bank no. which has the Error. <code>ltcd_lbcf_parity_err_banknum[1:0]</code>. Once set by HW, it can be cleared only by MMIO Write of 1 to this register bit 13. Driver needs to write 1 to clear this bit.</td>
</tr>
</tbody>
</table>
## L3CDERRST - L3CD Error Status register 1

<table>
<thead>
<tr>
<th>Bit(s)</th>
<th>Field Description</th>
<th>Default Value</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>10:8</td>
<td>Parity Error sub-bank no</td>
<td></td>
<td>R/W</td>
<td>One Clear</td>
</tr>
<tr>
<td></td>
<td>Parity Error in sub bank: ltcd0_lbcf_parErr_subBankNum[2:0]. Once set by HW, it can be cleared only by MMIO Write of 1 to this register bit 13. Driver needs to write 1 to clear this bit.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Parity report enable</td>
<td>1b</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Parity report enable (LCPRTYRPTEN): lbcf_csr_lc_parity_report_en. This is the parity reporting enable, by default it is enabled. When enabled parity is reported by ltcd to sarb. When disabled by driver, ltcd should not send out any parity error to SARB. Driver needs to write 1 to clear this bit.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6:0</td>
<td>Reserved</td>
<td></td>
<td>RO</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
L3 Control Register

L3CNTLREG - L3 Control Register

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000 CHV, BSW</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>07034h</td>
</tr>
</tbody>
</table>

**Programming Notes**

The L3 allocation programming should assign all ways of the cache with no left over ways. Refer to L3 section for the recommended settings.

Any L3 configuration change that reduces the data cache allocation when strong IA coherency is used requires the full flush of L3 prior to the programming update.

An explicit or implicit flush of L3 (DC Flush) through the command streamer doesn't result in flushing/invalidating the IA Coherent lines from L3. However this can be achieved by setting the "Pipe line flush Coherent lines" control bit in the "L3SQCREG4" register.

SLM comes up in an in-consistent state post reconfiguration and must be initialized by the driver for proper parity generation.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:25</td>
<td><strong>All L3 Client Pool</strong></td>
<td>CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: All</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Number of ways allocated for the all client pool. This is a combined pool for all clients.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>30h</td>
<td>[Default] CHV, BSW</td>
<td></td>
</tr>
</tbody>
</table>

**Programming Notes**

When this field is non-zero, DC Way Assignment and Read Only Client Pool should be 0KB. Odd number values are not allowed. Please refer to L3 Section with Allocation and Programming for recommended settings.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>24:18</td>
<td><strong>DC Way Assignment</strong></td>
<td>CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: All</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Number of ways allocated for DC. Note this allocation is only for DC data types.</td>
<td></td>
</tr>
</tbody>
</table>

**Programming Notes**

Note: This field must be 0KB if All L3 Client Pool is non-zero. Odd number values are not allowed. Please refer to L3 HAS for valid programming values.
# L3CNTLREG - L3 Control Register

## 17:11 Read Only Client Pool

<table>
<thead>
<tr>
<th>Project</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access</td>
<td>R/W</td>
</tr>
</tbody>
</table>

Number of ways allocated for Read Only L3 clients. This is a combined pool for all Read Only clients.

### Programming Notes

Note: This field must be 0KB if All L3 Client Pool is non-zero. Odd number values are not allowed. **Please refer to L3 HAS for valid programming values**

## 10 Reserved

<table>
<thead>
<tr>
<th>Access</th>
<th>R/W</th>
</tr>
</thead>
<tbody>
<tr>
<td>Format</td>
<td>PBC</td>
</tr>
</tbody>
</table>

## 9 Error Detection Behavior Control

<table>
<thead>
<tr>
<th>Project</th>
<th>CHV, BSW</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access</td>
<td>R/W</td>
</tr>
<tr>
<td>Format</td>
<td>Enable</td>
</tr>
</tbody>
</table>

The L3 error detection can be enabled to hang the GPU on a non-recoverable error due to SER type events. Such option will be used when corresponding context has data consistency requirements. Once error detection is enabled, s/w has to initialize URB or SLM to all 0’s (based on usage model) prior to execution of the workload. Initialization is required to clean up the error detection logic and syndrome tracking.

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
<th>Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>[Default]</td>
<td>RTL does not hang on parity errors or double bit error</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>1h</td>
<td></td>
<td>RTL enforces a hang on parity errors or double bit error</td>
<td>CHV, BSW</td>
</tr>
</tbody>
</table>

## 8 GPGPU L3 Credit Mode Enable

<table>
<thead>
<tr>
<th>Project</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access</td>
<td>R/W</td>
</tr>
<tr>
<td>Format</td>
<td>Enable</td>
</tr>
</tbody>
</table>

This bit is required to be enabled under GPGPU workloads to provide the MAX latency coverage from L3 cache. It will override the registers 0xB100[18:14] and 0xB100[23:19], to 0 and the maximum value respectively.

## 7:1 URB Allocation

<table>
<thead>
<tr>
<th>Project</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access</td>
<td>R/W</td>
</tr>
</tbody>
</table>

Number of ways allocated for URB usage

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>30h</td>
<td>[Default]</td>
<td>CHV, BSW</td>
</tr>
</tbody>
</table>

### Programming Notes

Odd number values are not allowed. **Please refer to L3 HAS for valid programming values**
<table>
<thead>
<tr>
<th>0</th>
<th><strong>SLM Mode Enable</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Project:</strong></td>
<td>All</td>
</tr>
<tr>
<td><strong>Access:</strong></td>
<td>R/W</td>
</tr>
<tr>
<td><strong>Format:</strong></td>
<td>Enable</td>
</tr>
</tbody>
</table>

When enabled, a 64KB (per bank) region of L3 is reserved for SLM.
# L3 Control Register1

## L3CNTLREG1 - L3 Control Register1

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Value</th>
<th>Name</th>
<th>Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:28</td>
<td><strong>Data Fifo Depth Control</strong></td>
<td>1000b</td>
<td>[Default]</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Data Fifo Depth Control (TS mode).</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Value cannot be zero for normal operation.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>lbcf_csr_lc_datafifo_depth[3:0].</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>27:24</td>
<td></td>
<td><strong>Data Clock off time</strong></td>
<td>1100b</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Data Clock off time (DATACLKOFF):</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Data Clock off time - Data block is shut off after these many number of clocks programmed in this register bits.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>lbcf_csr_lc_dataclkoff_time[3:0].Min value to be 4'h0100.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>It should be between 4'h4 : 4'hf.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23:20</td>
<td></td>
<td><strong>TAG CLK OFF TIME</strong></td>
<td>0100b</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td></td>
<td><strong>L3 Aging Disable Bit</strong></td>
<td>0b</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>L3 Aging Disable Bit (L3AGDIS): Aging Disable.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>lbcf_csr_lc_agingdis.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18:15</td>
<td></td>
<td><strong>Fill aging</strong></td>
<td>1111b</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value:</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# L3CNTLREG1 - L3 Control Register 1

<table>
<thead>
<tr>
<th>Field</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fill aging (L3AGF): Aging Counter for Fill.</td>
<td>R/W</td>
<td>If bit B103.19 is 0 then this register value has to be nonzero.</td>
</tr>
<tr>
<td>14:11 Aging Counter for Read 1 Port</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default Value:</td>
<td>1111b</td>
<td></td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>10:7 L3 Aging Counter for R0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default Value:</td>
<td>1111b</td>
<td></td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>6:0 Reserved</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default Value:</td>
<td>000000b</td>
<td></td>
</tr>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
<td></td>
</tr>
<tr>
<td>Access:</td>
<td>RO</td>
<td>Reserved.</td>
</tr>
</tbody>
</table>

- **Access**: R/W
- **Fill aging (L3AGF): Aging Counter for Fill.** If bit B103.19 is 0 then this register value has to be nonzero.
- **14:11 Aging Counter for Read 1 Port**
  - Default Value: 1111b
  - Access: R/W
  - Aging Counter for Read 1 Port (L3AGR1): Aging Counter for Read 1 Port. If bit B103.19 is 0 then this register value has to be nonzero.
- **10:7 L3 Aging Counter for R0**
  - Default Value: 1111b
  - Access: R/W
  - L3 Aging Counter for R0 (L3AGR0): Aging Counter for R0 Port. If bit B103.19 is 0 then this register value has to be nonzero.
- **6:0 Reserved**
  - Default Value: 000000b
  - Project: CHV, BSW
  - Access: RO
  - Reserved.
## L3 LRA 0

### L3_LRA_0 - L3 LRA 0

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:30</td>
<td><strong>L3</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Which LRA should L3 use.</td>
</tr>
<tr>
<td></td>
<td>29:20</td>
<td><strong>L3 LRA1 Min</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0011000010b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Minimum value of programmable LRA1.</td>
</tr>
<tr>
<td></td>
<td>19:10</td>
<td><strong>L3 LRA0 Max</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0011011111b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Maximum value of programmable LRA0.</td>
</tr>
<tr>
<td></td>
<td>9:0</td>
<td><strong>L3 LRA0 Min</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0000000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Minimum value of programmable LRA0.</td>
</tr>
</tbody>
</table>
**L3 LRA 0 GPGPU**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x0400FC00</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>04DD0h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:30</td>
<td><strong>L3 GPGPU</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Which LRA should L3 use.</td>
</tr>
<tr>
<td>29:20</td>
<td></td>
<td><strong>L3 LRA1 Min GPGPU</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0001000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Minimum value of programmable LRA1.</td>
</tr>
<tr>
<td>19:10</td>
<td></td>
<td><strong>L3 LRA0 Max GPGPU</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0000111111b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Maximum value of programmable LRA0.</td>
</tr>
<tr>
<td>9:0</td>
<td></td>
<td><strong>L3 LRA0 Min GPGPU</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0000000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Minimum value of programmable LRA0.</td>
</tr>
</tbody>
</table>
## L3 LRA 1

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x67F701BF</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Exists If:</td>
<td>Device[Platform] == 'Client'</td>
</tr>
<tr>
<td>Address:</td>
<td>04A14h</td>
</tr>
</tbody>
</table>

### Description

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:30</td>
<td><strong>DC</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 01b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Which LRA should DC use.</td>
</tr>
<tr>
<td>29:20</td>
<td></td>
<td><strong>L3 LRA2 Max</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 1001111111b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Maximum value of programmable LRA2.</td>
</tr>
<tr>
<td>19:10</td>
<td></td>
<td><strong>L3 LRA2 Min</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0111000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Minimum value of programmable LRA2.</td>
</tr>
<tr>
<td>9:0</td>
<td></td>
<td><strong>L3 LRA1 Max</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0110111111b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Maximum value of programmable LRA1.</td>
</tr>
</tbody>
</table>
## L3 LRA 1 GPGPU

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:30</td>
<td><strong>DC GPGPU</strong></td>
<td>01b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>29:20</td>
<td><strong>L3 LRA2 Max GPGPU</strong></td>
<td>0111111111b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>19:10</td>
<td><strong>L3 LRA2 Min GPGPU</strong></td>
<td>0110110000b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>9:0</td>
<td><strong>L3 LRA1 Max GPGPU</strong></td>
<td>0110101111b</td>
<td>R/W</td>
</tr>
</tbody>
</table>

**Register Space:** MMIO: 0/2/0  
**Project:** CHV, BSW  
**Source:** PRM  
**Default Value:** 0x5FF6C1AF  
**Size (in bits):** 32  
**Address:** 04DD4h
## L3 LRA 2

<table>
<thead>
<tr>
<th>Register Space</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000002</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Exists If:</td>
<td>Device[Platform] == 'Client'</td>
</tr>
<tr>
<td>Address:</td>
<td>04A18h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:2</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000000000000000000000000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>1:0</td>
<td></td>
<td><strong>Texture</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 10b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

Which LRA should Texture use.
## L3 LRA 2 GPGPU

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:2</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 000000000000000000000000000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>1:0</td>
<td></td>
<td><strong>Texture GPGPU</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 10b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

Which LRA should Texture use.
## L3 LRA 0 3D

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:30</td>
<td><strong>L3 3D</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Which LRA should L3 use.</td>
</tr>
<tr>
<td>0</td>
<td>29:20</td>
<td><strong>L3 LRA1 Min 3D</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0001000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Minimum value of programmable LRA1.</td>
</tr>
<tr>
<td>0</td>
<td>19:10</td>
<td><strong>L3 LRA0 Max 3D</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0000111111b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Maximum value of programmable LRA0.</td>
</tr>
<tr>
<td>0</td>
<td>9:0</td>
<td><strong>L3 LRA0 Min 3D</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0000000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Minimum value of programmable LRA0.</td>
</tr>
</tbody>
</table>
## L3 LRA 1 3D

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Value</th>
<th>Name</th>
<th>Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:30</td>
<td><strong>DC 3D</strong></td>
<td>01b</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access:</td>
<td>R/W</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Which LRA should DC use.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>29:20</td>
<td><strong>L3 LRA2 Max 3D</strong></td>
<td>R/W</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Maximum value of programmable LRA2.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Value</td>
<td>Name</td>
<td>Project</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0111111111b</td>
<td>[Default]</td>
<td>CHV, BSW</td>
<td></td>
</tr>
<tr>
<td></td>
<td>19:10</td>
<td><strong>L3 LRA2 Min 3D</strong></td>
<td>0010010000b</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access:</td>
<td>R/W</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Minimum value of programmable LRA2.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>9:0</td>
<td><strong>L3 LRA1 Max 3D</strong></td>
<td>0010001111b</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access:</td>
<td>R/W</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Maximum value of programmable LRA1.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### L3 LRA 2 3D

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:2</td>
<td>Reserved</td>
<td>000000000000000000000000000000b</td>
<td>RO</td>
</tr>
<tr>
<td>1:0</td>
<td></td>
<td>Texture 3D</td>
<td>10b</td>
<td>R/W</td>
</tr>
</tbody>
</table>

Which LRA should Texture use.
## L3 Messaging Register

### MSG_L3_LPFC - L3 Messaging Register

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>16</td>
</tr>
<tr>
<td>Address:</td>
<td>08038h</td>
</tr>
</tbody>
</table>

Message registers have bit-wise masking applied for writes. The register consists of 16 bits of data in [15:0], and 16 bits of corresponding masks in [31:16]. In order to set bit0, for example, the data would be 0x0001_0001. In order to clear bit0, for example, the data would be 0x0001_0000. Note that mask bit is the data bit offset + 16. Message registers are protected from non-GT writes via the Message Channel.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Access:</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>15:2</td>
<td>Reserved</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td><strong>Acknowledge that L3 Unblock Completed</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Acknowledge that L3 Unblock Completed</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1'b0 : L3 unblock not complete yet (default)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1'b1 : L3 unblock has completed</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>gpmunit self-clears this bit upon sampling.</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td><strong>Acknowledge that L3 Flush and Block Completed</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Acknowledge that L3 Flush and Block Completed</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1'b0 : L3 flush and block not complete yet (default)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1'b1 : L3 flush and block has completed</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>gpmunit self-clears this bit upon sampling.</td>
<td></td>
</tr>
</tbody>
</table>
## L3 SLM Register

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong></td>
<td>MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong></td>
<td>CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong></td>
<td>PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong></td>
<td>0x40000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong></td>
<td>32</td>
</tr>
<tr>
<td><strong>Address:</strong></td>
<td>0B110h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31</td>
<td><strong>Disable Periodic SLM/SQ slot allocation</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Disable Periodic SLM/SQ slot allocation: When <code>cfg_lslm_livelock_fairarb_dis=1</code> lslm unit always has the higher priority and lslm_lsqc_block to lsqcunit is asserted as long as there are requests in SLM FIFO. <code>lbcf_csr_lslm_livelock_fairarb_dis</code>.</td>
</tr>
<tr>
<td>30:26</td>
<td></td>
<td><strong>LSLM_SQ_PENDING_MAX</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 10000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If lslmunit has read data to be sent to lcbrunit this cfg register specifies the maximum number of clocks for which LSLMunit can block SQ request from being sent o lcbrunit.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default value = 8.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Value cannot be zero.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>lbcf_csr_lslm_sqpend_max[4:0]</code>.</td>
</tr>
<tr>
<td>25</td>
<td></td>
<td><strong>LSLM address disable</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 - Enable b2b addr matching fix. Islmunit should not block the cycle in fifo if there is a match in the pipeline.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 - Disable b2b addr matching fix. Islmunit should block the cycle in fifo if there is a match in the pipeline.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>lbcf_csr_lslm_same_addr_dis</code>.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default = 0.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Set this bit to 1'b1 to workaround Atomic b2b bug on SLM for CHV, BSW A-step only.</td>
</tr>
<tr>
<td>24:0</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
</tbody>
</table>
### L3 SQC register 4

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x40400000 CHV, BSW</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0B118h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td>30</td>
<td>31</td>
<td><strong>L3SQ URB Read CAM Match Disable</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 1b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>L3SQ URB Read CAM Match Disable (SQRBRDCAMDIS):</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Disables the L3SQ Cam Match ability for URB Reads. By disabling, this allows a performance mode where URB reads are not dependent upon one another but only on any previous URB writes to the same address. This allows many URB reads to the same cacheline at any given time instead of serializing the requests. 1 = URB Read CAM matching is disabled; multiple URB reads to the same cacheline are allowed to be concurrent (default). 0 = URB Read CAM matching is enabled; multiple URB reads to the same cacheline are serialized. lbcf_csr_lsqc_urbrdcam_dis.</td>
</tr>
<tr>
<td>29:28</td>
<td></td>
<td><strong>Traffic regulation in LSQC for URB lookup traffic</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Traffic regulation in LSQC for URB lookup traffic (URB lookups are issued to ltcc these many clocks apart). 00b - Continuous. 01b - 4 clocks apart. 10b - 8 clocks apart. 11b - 16 clocks apart. lbcf_lsqc_urb_traffic.</td>
</tr>
<tr>
<td>27</td>
<td></td>
<td><strong>LQSC RO PERF DIS</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default: 0. when set, RO performance mode is disabled and all Reads proceed only after Parent recycles. lbcf_csr_lsqc_roperf_dis.</td>
</tr>
<tr>
<td>Register</td>
<td>Name</td>
<td>Default Value</td>
</tr>
<tr>
<td>----------</td>
<td>------</td>
<td>---------------</td>
</tr>
<tr>
<td>26</td>
<td>Order Cam Snp Reject</td>
<td>0b</td>
</tr>
<tr>
<td>25</td>
<td>LQSC RW PERF DIS</td>
<td>0b</td>
</tr>
<tr>
<td>24</td>
<td>LSQC read rtrn local crdt pre-consume disable</td>
<td>0b</td>
</tr>
<tr>
<td>23</td>
<td>LSQC Mem Write sqcam HITM response disable</td>
<td>0b</td>
</tr>
<tr>
<td>22</td>
<td>Non-IA coherent atomics enable</td>
<td>1b</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td><strong>Pipe line flush Coherent lines</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default Value: 0b</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Project: CHV, BSW</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1: Treat pipeline flush as invalidating even coherent lines along with non coherent lines . 0: Flush invalidates non coherent lines only.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>lbcf_csr_lsqc_pipeflush_coh.</td>
<td></td>
</tr>
<tr>
<td>20:0</td>
<td><strong>Reserved</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Project: CHV, BSW</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access: RO</td>
<td></td>
</tr>
</tbody>
</table>
### L3SQCREG1 - L3 SQC registers 1

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td>Reserved.</td>
<td></td>
</tr>
<tr>
<td>23:19</td>
<td><strong>L3SQ General Priority Credit Initialization</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>R/W</td>
</tr>
</tbody>
</table>

**Description**

L3SQ General Priority Credit Initialization (SQGPCI):
Number of general and high priority credits that SQ presents to L3 Arbiter blocks. This inherently also determines the depth of the SQ; reduce the number of credits and SQ uses fewer slots.

Any value not listed here is considered Reserved.
Gen priority credits is always greater than high priority credits.
When this register is programmed through KMD, the crclk DOP clk gating should be disabled before the programming and be enabled ~100 clocks after the programming is done.

Value

<table>
<thead>
<tr>
<th># General Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000b</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>00001b</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>00010b</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>00011b</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>00100b</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>00101b</td>
</tr>
<tr>
<td>10</td>
</tr>
<tr>
<td>00110b</td>
</tr>
<tr>
<td>12</td>
</tr>
<tr>
<td>00111b</td>
</tr>
</tbody>
</table>
## L3SQCREG1 - L3 SQC registers

### Values and Description

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>01000b</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>01001b</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>01010b</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>01011b</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>01100b</td>
<td></td>
</tr>
<tr>
<td>24 (default)</td>
<td>01101b</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>01110b</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>01111b</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>10000b</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td></td>
<td></td>
</tr>
<tr>
<td>34</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100010</td>
<td></td>
<td></td>
</tr>
<tr>
<td>36</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10011</td>
<td></td>
<td></td>
</tr>
<tr>
<td>38</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Other values are not possible.

Need to go up to 32 credits.

The value of `lbcf_csr_lsqc_gen_credit_init[4:0]`.

### 18:14 L3SQ High Priority Credit Initialization

<table>
<thead>
<tr>
<th>Default Value:</th>
<th>00100b</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
</tbody>
</table>

**L3SQ High Priority Credit Initialization (SQHPCI):**

Number of general and high priority credits that SQ presents to L3 Arbiter blocks. This inherently also determines the depth of the SQ; reduce the number of credits and SQ uses fewer slots.

Any value not listed here is considered Reserved.

The number of general priority credits is always greater than high priority credits.
### L3SQCREG1 - L3 SQC registers 1

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>When this register is programmed through KMD, the crclk DOP clk gating should be disabled before the programming and be enabled ~100 clocks after the programming is done.</strong>&lt;br&gt;<strong>Value</strong>&lt;br&gt;# High Pri Credits&lt;br&gt;00000b 0&lt;br&gt;00001b 2&lt;br&gt;00010b 4&lt;br&gt;00011b 6&lt;br&gt;00100b 8 (default)&lt;br&gt;00101b 10&lt;br&gt;00110b 12&lt;br&gt;00111b 14&lt;br&gt;01000b 16&lt;br&gt;01001b 18&lt;br&gt;01010b 20&lt;br&gt;01011b 22&lt;br&gt;01100b 24&lt;br&gt;01101b 26&lt;br&gt;01110b 28&lt;br&gt;01111b 30&lt;br&gt;10000b 32&lt;br&gt;Other values are not possible. lbcf_csr_lsqc_hp_credit_init[4:0] lbcf_csr_lsqc_gen_credit_init[4:0] should always be less than or equal to 32.</td>
<td></td>
</tr>
<tr>
<td>13:10 <strong>Reserved</strong></td>
<td>Access: RO Reserved.</td>
</tr>
</tbody>
</table>
### L3SQCREG1 - L3 SQC registers 1

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
</tr>
</thead>
</table>
| **9** L3SQ Read Once Enable for Sampler Client | Access: R/W  
L3SQ Read Once Enable for Sampler Client (SQROE): Enables Read Once indications to L3 Cache from SQ. Once enabled, any reads from Sampler client (MT) are sent as Read Once.  
0 = (default) Reads from Sampler clients issue Read to L3 Cache.  
1 = Reads from Sampler clients issue Read Once to L3 Cache.  
lbcf_csr_sampler_readonce_en. |
| **8:6** Reserved | Access: RO  
Reserved. |
| **5:3** L3SQ Outstanding L3 Fills | Access: R/W  
L3SQ Outstanding L3 Fills (SQOUTSL3F): Identifies the number of L3 Fills that can be outstanding before SQ throttles the fill requests to L3 Cache. This is not an exact limit, but instead it is used as a threshold to throttling. Once the fill count is greater than or equal to the threshold, then no fills are issued until the fill responses are received to bring the outstanding count back below the threshold.  
000b = (default) No limit.  
001b = 1 fill.  
010b = 2 fills.  
011b = 4 fills.  
100b = 8 fills.  
101b = 16 fills.  
11Xb = Reserved.  
lbcf_csr_lsqc_outs_fill[2:0]. |
| **2:0** L3SQ Outstanding L3 Lookups | Access: R/W  
L3SQ Outstanding L3 Lookups (SQOUTSL3L): Identifies the number of L3 lookups that can be outstanding before SQ throttles the lookup requests to L3 Cache. This is not an exact limit, but instead it is used as a threshold to throttling. Once the lookup count is greater than or equal to the threshold, then no lookups are issued until the lookup responses are received to bring the outstanding count back below the threshold.  
000b = (default) No limit.  
001b = 1 lookup.  
010b = 2 lookups.  
011b = 4 lookups.  
100b = 8 lookups.  
101b = 16 lookups.  
11Xb = Reserved.  
lbcf_csr_lsqc_outs_lookup[2:0]. |
# L3 SQC registers 2

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00004567</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0B104h</td>
</tr>
</tbody>
</table>

### Register: L3SQCREG2 - L3 SQC registers 2

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:17 Reserved</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access: RO</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Reserved.</td>
<td></td>
</tr>
</tbody>
</table>

**Description**

L3SQ Priority Selection Disable (SQPRIDIS):
Enables the use of priority selection based on client ID decodes. If disabled, all cycles in SQ are treated as same priority.
0 = (default) Priority selection is enabled.
1 = Priority selection is disabled.
Should not be set when RO perf mode is enabled (by default this is enabled which is bit[27] of B11Bh). lbcf_csr_priority_cnt_disable.

**Workaround:** If this bit is set to 1 then the following bits in register B108-B10B should have identical values 29:28, 27:26, 25:24, 23:22, 21:20, 19:18, 17:16, 15:14, 13:12, 11:10, 9:8, 7:6, 5:4, 3:2, 1:0.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td></td>
<td>L3SQ Priority Selection Disable</td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
<td></td>
</tr>
</tbody>
</table>

**Description**

L3SQ Priority Selection Disable (SQPRIDIS):
When set, priority3 pool becomes unlimited. And priority3 pool count value should not be used in reset of the remaining counters.
0 = (default) Priority 3 pool count is enabled.
1 = Priority 3 pool count is disabled.
Should not be set when RO perf mode is enabled (by default this is enabled which is bit[27] of B11Bh). lbcf_csr_priority3_cnt_disable.
<table>
<thead>
<tr>
<th>14:12</th>
<th>L3SQ Priority 3 Pool Counter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Value:</td>
<td>100b</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>L3SQ Priority 3 Pool Counter (SQPRI3CNT):</td>
<td></td>
</tr>
<tr>
<td>The count of cycles is selected from priority3 pool before switching to other priority pools. Count is used as the power of 2.</td>
<td></td>
</tr>
<tr>
<td>000b = 1 request.</td>
<td></td>
</tr>
<tr>
<td>001b = 2 requests.</td>
<td></td>
</tr>
<tr>
<td>010b = 4 requests.</td>
<td></td>
</tr>
<tr>
<td>011b = 8 requests.</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>111b = 128 requests.</td>
<td></td>
</tr>
<tr>
<td>lbcf_csr_priority3_cnt[2:0].</td>
<td></td>
</tr>
</tbody>
</table>

| 11 | L3SQ Priority 2 Pool Count Disable |
| Access: | R/W |
| Description |
| L3SQ Priority 2 Pool Count Disable (SQPRI2CNTDIS): |
| When set, priority2 pool becomes unlimited. And priority2 pool count value should not be used in reset of the remaining counters. |
| 0 = (default) Priority 2 pool count is enabled. |
| 1 = Priority 2 pool count is disabled. |
| Should not be set when RO perf mode is enabled (by default this is enabled which is bit[27] of B118-B11Bh). |
| lbcf_csr_priority2_cnt_disable. |
| Workaround: If this bit is set to 1 then the following bits in register B108-B10B should have identical values 29:28, 27:26, 25:24, 23:22, 21:20, 19:18, 17:16, 15:14, 13:12, 11:10, 9:8, 7:6, 5:4, 3:2, 1:0. |

| 10:8 | L3SQ Priority 2 Pool Counter |
| Default Value: | 101b |
| Access: | R/W |
| L3SQ Priority 2 Pool Counter (SQPRI2CNT): |
| The count of cycles is selected from priority2 pool before switching to other priority pools. Count is used as the power of 2. |
| 000b = 1 request. |
| 001b = 2 requests. |
| 010b = 4 requests. |
| 011b = 8 requests. |
| ... |
| 111b = 128 requests. |
### L3SQCREG2 - L3 SQC registers 2

<table>
<thead>
<tr>
<th>7</th>
<th><strong>L3SQ Priority 1 Pool Count Disable</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
</tbody>
</table>

**Description**

L3SQ Priority 1 Pool Count Disable (SQPRI1CNTDIS):
When set, priority1 pool becomes unlimited. And priority1 pool count value should not be used in reset of the remaining counters.
0 = (default) Priority 1 pool count is enabled.
1 = Priority 1 pool count is disabled.
Should not be set when RO perf mode is enabled (by default this is enabled which is bit[27] of B118-B11Bh).
lbcf_csr_priority1_cnt_disable.

Workaround: If this bit is set to 1 then the following bits in register B108-B10B should have identical values 29:28, 27:26, 25:24, 23:22, 21:20, 19:18, 17:16, 15:14, 13:12, 11:10, 9:8, 7:6, 5:4, 3:2, 1:0.

<table>
<thead>
<tr>
<th>6:4</th>
<th><strong>L3SQ Priority 1 Pool Counter</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Value:</td>
<td>110b</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
</tbody>
</table>

L3SQ Priority 1 Pool Counter (SQPRI1CNT):
The count of cycles is selected from priority1 pool before switching to other priority pools. Count is used as the power of 2.
000b = 1 request.
001b = 2 requests.
010b = 4 requests.
011b = 8 requests.
...
111b = 128 requests.
lbcf_csr_priority1_cnt[2:0].

<table>
<thead>
<tr>
<th>3</th>
<th><strong>L3SQ Priority 0 Pool Count Disable</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
</tbody>
</table>

**Description**

L3SQ Priority 0 Pool Count Disable (SQPRI0CNTDIS):
When set, priority0 pool becomes unlimited. And priority0 pool count value should not be used in reset of the remaining counters.
0 = (default) Priority 0 pool count is enabled.
1 = Priority 0 pool count is disabled.
Should not be set when RO perf mode is enabled (by default this is enabled which is bit[27] of B118-B11Bh).
lbcf_csr_priority0_cnt_disable.
## L3SQCREG2 - L3 SQC registers 2

<table>
<thead>
<tr>
<th>2:0</th>
<th><strong>L3SQ Priority 0 Pool Counter</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Default Value: 111b</td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

L3SQ Priority 0 Pool Counter (SQPRI0CNT):
The count of cycles is selected from priority0 pool before switching to other priority pools. Count is used as the power of 2.
- 000b = 1 request.
- 001b = 2 requests.
- 010b = 4 requests.
- 011b = 8 requests.
- ...
- 111b = (default) 128 requests.

lbcr_csr_priority0_cnt[2:0].
### L3SQCREG3 - L3 SQC registers 3

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00001ABF</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0B108h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:30</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved.</td>
</tr>
<tr>
<td>29:28</td>
<td></td>
<td><strong>SOLunit Priority Value</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SOLunit Priority Value (SQSOLPRIVAL):</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Identifies the priority value for all cycles</td>
</tr>
<tr>
<td></td>
<td></td>
<td>that are initiated by SOLunit. Priority is</td>
</tr>
<tr>
<td></td>
<td></td>
<td>used in the L3 Super Queue (L3SQ).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>00b = Priority 0 (default).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>01b = Priority 1.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10b = Priority 2.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>11b = Priority 3.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>lbcf_csr_sol_priority[1:0].</td>
</tr>
<tr>
<td>27:26</td>
<td></td>
<td><strong>GSunit Priority Value</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GSunit Priority Value (SQGSPRIVAL):</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Identifies the priority value for all cycles</td>
</tr>
<tr>
<td></td>
<td></td>
<td>that are initiated by GSunit. Priority is</td>
</tr>
<tr>
<td></td>
<td></td>
<td>used in the L3 Super Queue (L3SQ).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>00b = Priority 0 (default).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>01b = Priority 1.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10b = Priority 2.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>11b = Priority 3.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>lbcf_csr_gs_priority[1:0].</td>
</tr>
<tr>
<td>25:24</td>
<td></td>
<td><strong>TEunit Priority Value</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TEunit Priority Value (SQTEPRIVAL):</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Identifies the priority value for all cycles</td>
</tr>
<tr>
<td></td>
<td></td>
<td>that are initiated by TEunit. Priority is</td>
</tr>
<tr>
<td></td>
<td></td>
<td>used in the L3 Super Queue (L3SQ).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>00b = Priority 0 (default).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>01b = Priority 1.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10b = Priority 2.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>11b = Priority 3.</td>
</tr>
</tbody>
</table>
## L3SQCREG3 - L3 SQC registers 3

<table>
<thead>
<tr>
<th>Address</th>
<th>Name</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>23:22</td>
<td>L3 Clunit Priority Value</td>
<td>R/W</td>
<td>CLunit Priority Value (SQCLPRIVAL): Identifies the priority value for all cycles that are initiated by CLunit. Priority is used in the L3 Super Queue (L3SQ). 00b = Priority 0 (default). 01b = Priority 1. 10b = Priority 2. 11b = Priority 3.</td>
</tr>
<tr>
<td>21:20</td>
<td>L3 TSunit Priority Value</td>
<td>R/W</td>
<td>TSunit Priority Value (SQTSPRIVAL): Identifies the priority value for all cycles that are initiated by TSunit. Priority is used in the L3 Super Queue (L3SQ). 00b = Priority 0 (default). 01b = Priority 1. 10b = Priority 2. 11b = Priority 3.</td>
</tr>
<tr>
<td>19:18</td>
<td>L3 SFunit Priority Value</td>
<td>R/W</td>
<td>SFunit Priority Value (SQSFPRIVAL): Identifies the priority value for all cycles that are initiated by SFunit. Priority is used in the L3 Super Queue (L3SQ). 00b = Priority 0 (default). 01b = Priority 1. 10b = Priority 2. 11b = Priority 3.</td>
</tr>
<tr>
<td>17:16</td>
<td>L3 SVSM Priority Value</td>
<td>R/W</td>
<td>SVSM Priority Value (SQSVSMPRIVAL): Identifies the priority value for all cycles that are initiated by SVSM. Priority is used in the L3 Super Queue (L3SQ). 00b = Priority 0 (default). 01b = Priority 1. 10b = Priority 2. 11b = Priority 3.</td>
</tr>
<tr>
<td>15:14</td>
<td>L3 SARB Priority Value</td>
<td>R/W</td>
<td>SARB Priority Value: Identifies the priority value for all cycles that are initiated by SARB. Priority is used in the L3 Super Queue (L3SQ). 00b = Priority 0 (default). 01b = Priority 1. 10b = Priority 2. 11b = Priority 3.</td>
</tr>
</tbody>
</table>
### L3SQCREG3 - L3 SQC registers 3

<table>
<thead>
<tr>
<th>Field</th>
<th>Default Value</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SARB Priority Value (SQSARBPRIVAL):</strong></td>
<td></td>
<td></td>
<td>Identifies the priority value for all cycles that are initiated by State Arbiter (SARB). Priority is used in the L3 Super Queue (L3SQ).</td>
</tr>
<tr>
<td>00b</td>
<td>01b</td>
<td></td>
<td>Priority 0 (default).</td>
</tr>
<tr>
<td>01b</td>
<td>01b</td>
<td></td>
<td>Priority 1 (default).</td>
</tr>
<tr>
<td>10b</td>
<td>01b</td>
<td></td>
<td>Priority 2.</td>
</tr>
<tr>
<td>11b</td>
<td>01b</td>
<td></td>
<td>Priority 3.</td>
</tr>
<tr>
<td>lbcf_csr_sarb_priority[1:0].</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>SBE Priority Value</strong></td>
<td></td>
<td></td>
<td>Identifies the priority value for all cycles that are initiated by SBE. Priority is used in the L3 Super Queue (L3SQ).</td>
</tr>
<tr>
<td>Default Value:</td>
<td>01b</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>Access:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lbcf_csr_sbe_priority[1:0].</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>ICS Priority Value</strong></td>
<td></td>
<td></td>
<td>Identifies the priority value for all cycles that are initiated by Instruction Cache (IC$). Priority is used in the L3 Super Queue (L3SQ).</td>
</tr>
<tr>
<td>Default Value:</td>
<td>10b</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>Access:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lbcf_csr_ic_priority[1:0].</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>TDL Priority Value</strong></td>
<td></td>
<td></td>
<td>Identifies the priority value for all cycles that are initiated by TDL. Priority is used in the L3 Super Queue (L3SQ).</td>
</tr>
<tr>
<td>Default Value:</td>
<td>10b</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>Access:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lbcf_csr_tdl_priority[1:0].</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>DCunit Priority Value</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default Value:</td>
<td>10b</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Access:</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## L3SQCREG3 - L3 SQC registers 3

<table>
<thead>
<tr>
<th>Access:</th>
<th>R/W</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCunit Priority Value (SQDCPRIVAL):</td>
<td>Identifies the priority value for all cycles that are initiated by DC. Priority is used in the L3 Super Queue (L3SQ). 00b = Priority 0. 01b = Priority 1. 10b = Priority 2 (default). 11b = Priority 3. <code>lbcf_csr_dc_priority[1:0]</code>.</td>
</tr>
<tr>
<td>DAPR Priority Value</td>
<td></td>
</tr>
<tr>
<td>Default Value:</td>
<td>11b</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>DAPR Priority Value (SQDAPRPRIVAL):</td>
<td>Identifies the priority value for all cycles that are initiated by DAPR. Priority is used in the L3 Super Queue (L3SQ). 00b = Priority 0. 01b = Priority 1. 10b = Priority 2. 11b = Priority 3 (default). <code>lbcf_csr_dapr_priority[1:0]</code>.</td>
</tr>
<tr>
<td>MTunit Priority Value</td>
<td></td>
</tr>
<tr>
<td>Default Value:</td>
<td>11b</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>MTunit Priority Value (SQMTPRIVAL):</td>
<td>Identifies the priority value for all cycles that are initiated by Sampler (MT). Priority is used in the L3 Super Queue (L3SQ). 00b = Priority 0. 01b = Priority 1. 10b = Priority 2. 11b = Priority 3 (default). <code>lbcf_csr_mt_priority[1:0]</code>.</td>
</tr>
<tr>
<td>LSQCunit Priority Value</td>
<td></td>
</tr>
<tr>
<td>Default Value:</td>
<td>11b</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>LSQCunit Priority Value (SQPRIVAL):</td>
<td>Identifies the priority value for all cycles that are initiated by Super Queue (L3 Evictions). Priority is used in the L3 Super Queue (L3SQ). 00b = Priority 0. 01b = Priority 1. 10b = Priority 2. 11b = Priority 3 (default). <code>lbcf_csr_lsqc_priority[1:0]</code>.</td>
</tr>
</tbody>
</table>
LBCF config save msg

<table>
<thead>
<tr>
<th>Register Space: MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 0B2FCh</td>
</tr>
</tbody>
</table>

This register is not context saved and is written by PM unit.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:10</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td>9:0</td>
<td><strong>Context save bit</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Hardware Clear</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>Bit[9]: Power Context Save Request</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0: Power context save is not being requested</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1: Power context save is being requested</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Unit needs to self-clear this bit upon sampling</td>
</tr>
<tr>
<td></td>
<td>8:0</td>
<td>Bits[8:0]: QWord Credits for Power Context</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Save Request</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Minimum Credits = 1: Unit may send 1 QWord</td>
</tr>
<tr>
<td></td>
<td></td>
<td>pair (enough for first LRI at least).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Maximum Credits = 511: Unit may send 511 QWord</td>
</tr>
<tr>
<td></td>
<td></td>
<td>pairs.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A QWord pair is defined as a 32-bit register</td>
</tr>
<tr>
<td></td>
<td></td>
<td>address and the corresponding 32-bits of</td>
</tr>
<tr>
<td></td>
<td></td>
<td>register data. Note that the LRI header and</td>
</tr>
<tr>
<td></td>
<td></td>
<td>END commands are 64-bits each (32-bit command</td>
</tr>
<tr>
<td></td>
<td></td>
<td>followed by 32-bit NOOP) and consumes one</td>
</tr>
<tr>
<td></td>
<td></td>
<td>QWord credit. Only valid with PWRCTX_SAVE_REQ</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Bit9).</td>
</tr>
</tbody>
</table>
## LBCF DPF Error log register 0

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LBCFPM00</td>
<td>LBCF DPF Error log register 0</td>
</tr>
</tbody>
</table>

### Register Space
- MMIO: 0/2/0

### Project
- CHV, BSW

### Source
- PRM

### Default Value
- 0x00000000

### Size (in bits)
- 32

### Address
- 0B130h

#### Slice0 Bank 0 subbank0 Error log register

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:21</td>
<td><strong>Row Number for Error 1</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Row Number for Error1: The physical row Addr where the parity error has been detected in SLM Bank. lbcf_sb0_error_addr1[9:0].</td>
</tr>
<tr>
<td>20:17</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td><strong>Valid Error 1</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Error: The Address located in field 31:21 is valid. lbcf_sb0_valid_error1.</td>
</tr>
<tr>
<td>15:5</td>
<td></td>
<td><strong>Row Number for Error0</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Row Number for Error0: The physical row Addr where the parity error has been detected in SLM Bank. lbcf_sb0_error_addr0[9:0].</td>
</tr>
<tr>
<td>4:1</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td><strong>Valid Error 0</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Error: The address located in field 15:5 is valid. lbcf_sb0_valid_error0.</td>
</tr>
</tbody>
</table>
## LBCF DPF Error log register 1

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:21</td>
<td><strong>Row Number for Error 1</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Row Number for Error1: The physical row Addr where the parity error has been detected in SLM Bank.</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>lbcf_sb1_error_addr1[9:0].</td>
<td></td>
</tr>
<tr>
<td></td>
<td>20:17</td>
<td><strong>Reserved</strong></td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td><strong>Valid Error 1</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Valid Error: The Address located in field 31:21 is valid.</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>lbcf_sb1_valid_error0.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>15:5</td>
<td><strong>Row Number for Error0</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Row Number for Error0: The physical row Addr where the parity error has been detected in SLM Bank.</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>lbcf_sb1_error_addr0[9:0].</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4:1</td>
<td><strong>Reserved</strong></td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td><strong>Valid Error 0</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Valid Error: The address located in field 15:5 is valid.</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>lbcf_sb1_valid_error0.</td>
<td></td>
</tr>
</tbody>
</table>
# LBCF DPF Error log register 2

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:21</td>
<td><strong>Row Number for Error 1</strong>&lt;br&gt;Access: <strong>R/W</strong>&lt;br&gt;Row Number for Error1: The physical row Addr where the parity error has been detected in SLM Bank. lbcf_sb2_error_addr1[9:0].</td>
</tr>
<tr>
<td></td>
<td>20:17</td>
<td><strong>Reserved</strong>&lt;br&gt;Access: <strong>RO</strong></td>
</tr>
<tr>
<td>16</td>
<td></td>
<td><strong>Valid Error 1</strong>&lt;br&gt;Access: <strong>R/W</strong>&lt;br&gt;Valid Error: The Address located in field 31:21 is valid. lbcf_sb2_valid_error1.</td>
</tr>
<tr>
<td>15:5</td>
<td></td>
<td><strong>Row Number for Error0</strong>&lt;br&gt;Access: <strong>R/W</strong>&lt;br&gt;Row Number for Error0: The physical row Addr where the parity error has been detected in SLM Bank. lbcf_sb2_error_addr0[9:0].</td>
</tr>
<tr>
<td>4:1</td>
<td></td>
<td><strong>Reserved</strong>&lt;br&gt;Access: <strong>RO</strong></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td><strong>Valid Error 0</strong>&lt;br&gt;Access: <strong>R/W</strong>&lt;br&gt;Valid Error: The address located in field 15:5 is valid. lbcf_sb2_valid_error0.</td>
</tr>
</tbody>
</table>
# LBCF DPF Error log register 3

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
</table>
| 0     | 31:21 | **Row Number for Error 1**  
Row Number for Error1: The physical row Addr where the parity error has been detected in SLM Bank.  
lbcf_sb3_error_addr1[9:0]. |
| 20:17 | Reserved |  
Access: RO |
| 16    | Valid Error 1  
Valid Error: The Address located in field 31:21 is valid.  
lbcf_sb3_valid_error1. |
| 15:5  | **Row Number for Error0**  
Row Number for Error0: The physical row Addr where the parity error has been detected in SLM Bank.  
lbcf_sb3_error_addr0[9:0]. |
| 4:1   | Reserved |  
Access: RO |
| 0     | Valid Error 0  
Valid Error: The address located in field 15:5 is valid.  
lbcf_sb3_valid_error0. |
## LBCF DPF Error log register 4

### LBCFERRLOG01 - LBCF DPF Error log register 4

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0B140h</td>
</tr>
</tbody>
</table>

Slice 0 Bank 1 Subbank0 Error log register

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:21</td>
<td><strong>Row Number for Error 1</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Row Number for Error 1: The physical row Addr where the parity error has been detected in SLM Bank.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Slice 0 Bank 1 Error log register 00.</td>
<td></td>
</tr>
<tr>
<td>20:17</td>
<td></td>
<td><strong>Reserved</strong></td>
<td>RO</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td><strong>Valid Error 1</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Error: The error located in field 31:21 is valid.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Slice 0 Bank 1 Error log register 00.</td>
<td></td>
</tr>
<tr>
<td>15:5</td>
<td></td>
<td><strong>Row Number for Error 0</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Row Number for Error 0: The physical row Addr where the parity error has been detected in SLM Bank.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Slice 0 Bank 1 Error log register 00.</td>
<td></td>
</tr>
<tr>
<td>4:1</td>
<td></td>
<td><strong>Reserved</strong></td>
<td>RO</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td><strong>Valid Error 0</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Error: The error located in field 15:5 is valid.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Slice 0 Bank 1 Error log register 00.</td>
<td></td>
</tr>
</tbody>
</table>
# LBCF DPF Error log register 5

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:21</td>
<td><strong>Row Number for Error 1</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Row Number for Error1: The physical row Addr where the parity error has been detected in SLM Bank. Slice 0 Bank 1 Error log register 00.</td>
</tr>
<tr>
<td>0</td>
<td>20:17</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td>0</td>
<td>15:5</td>
<td><strong>Row Number for Error0</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Row Number for Error0: The physical row Addr where the parity error has been detected in SLM Bank. Slice 0 Bank 1 Error log register 00.</td>
</tr>
<tr>
<td>0</td>
<td>4:1</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td><strong>Valid Error 0</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Error: The address located in field 15:5 is valid</td>
</tr>
</tbody>
</table>

Register Space: MMIO: 0/2/0  
Project: CHV, BSW  
Source: PRM  
Default Value: 0x00000000  
Size (in bits): 32  
Address: 0B144h  
Slice 0 Bank 1 Subbank1 Error log register
## LBCF DPF Error log register 6

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:21</td>
<td><strong>Row Number for Error 1</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Row Number for Error1: The physical row Addr where the parity error has been detected in SLM Bank.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Slice 0 Bank 1 Error log register 00.</td>
</tr>
<tr>
<td>20:17</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>RO</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td><strong>Valid Error 1</strong></td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Row Number for Error1: The physical row Addr where the parity error has been detected in SLM Bank.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Slice 0 Bank 1 Error log register 00.</td>
</tr>
<tr>
<td>15:5</td>
<td></td>
<td><strong>Row Number for Error0</strong></td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Row Number for Error0: The physical row Addr where the parity error has been detected in SLM Bank.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Slice 0 Bank 1 Error log register 00.</td>
</tr>
<tr>
<td>4:1</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>RO</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td><strong>Valid Error 0</strong></td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Error: The address located in field 15:5 is valid.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Slice 0 Bank 1 Error log register 00.</td>
</tr>
</tbody>
</table>
# LBCF DPF Error log register 7

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:21</td>
<td><strong>Row Number for Error 1</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Row Number for Error 1: The physical row Addr where the parity error has been detected in SLM Bank. Slice 0 Bank 1 Error log register 00.</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>15:5</td>
<td><strong>Row Number for Error 0</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Row Number for Error 0: The physical row Addr where the parity error has been detected in SLM Bank. Slice 0 Bank 1 Error log register 00.</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>4:1</td>
<td><strong>Reserved</strong></td>
<td>RO</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td><strong>Valid Error 0</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Error: The address located in field 15:5 is valid. Slice 0 Bank 1 Error log register 00.</td>
<td></td>
</tr>
</tbody>
</table>
# LBCF DPF Error log register 8

<table>
<thead>
<tr>
<th>LBCFERRLOG05 - LBCF DPF Error log register 8</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 0B150h</td>
</tr>
</tbody>
</table>

## Slice 0 Bank 2 Subbank0 Error log register

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:21</td>
<td><strong>Row Number for Error 1</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
</tbody>
</table>

Row Number for Error 1: The physical row Addr where the parity error has been detected in SLM Bank. The number of rows varies between 4K vs. 8K/16K subbanks which requires 10 bits vs 11 bits respectively. This field contains the row# with the error. Slice 0 Bank 2 Error log register 00.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>20:17</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> RO</td>
</tr>
</tbody>
</table>

16 | **Valid Error 1** |
| Access: | R/W |

Valid Error: The error located in field 16:5 is valid and Slice 0 Bank 2 Error log register 00.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>16:5</td>
<td></td>
<td><strong>Row Number for Error 0</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
</tbody>
</table>

Valid Error: The error located in field 16:5 is valid and Slice 0 Bank 2 Error log register 00.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4:1</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> RO</td>
</tr>
</tbody>
</table>

0 | **Valid Error 0** |
| Access: | R/W |

Valid Error: The error located in field 16:5 is valid. Slice 0 Bank 2 Error log register 00.
## LBCF DPF Error log register 9

<table>
<thead>
<tr>
<th><strong>LBCFERRLOG06 - LBCF DPF Error log register 9</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 0B154h</td>
</tr>
</tbody>
</table>

### Slice 0 Bank 2 subbank1 Error log register

<table>
<thead>
<tr>
<th><strong>DWORD</strong></th>
<th><strong>Bit</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>31:21</strong></td>
<td><strong>Row Number for Error 1</strong></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td>Row Number for Error1: The physical row Addr where the parity error has been detected in SLM Bank. The number of rows varies between 4K vs. 8K/16K subbanks which requires 10 bits vs 11 bits respectively. This field contains the row# with the error. Slice 0 Bank 2 Error log register 00.</td>
<td></td>
</tr>
<tr>
<td><strong>20:17</strong></td>
<td><strong>Reserved</strong></td>
<td>Access: RO</td>
</tr>
<tr>
<td><strong>16</strong></td>
<td><strong>Valid Error 1</strong></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td>Valid Error: The error located in field 31:21 is valid. Slice 0 Bank 2 Error log register 00.</td>
<td></td>
</tr>
<tr>
<td><strong>15:5</strong></td>
<td><strong>Row Number for Error0</strong></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td>Row Number for Error0: The physical row Addr where the parity error has been detected in SLM Bank. Slice 0 Bank 2 Error log register 00.</td>
<td></td>
</tr>
<tr>
<td><strong>4:1</strong></td>
<td><strong>Reserved</strong></td>
<td>Access: RO</td>
</tr>
<tr>
<td><strong>0</strong></td>
<td><strong>Valid Error 0</strong></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td>Valid Error: The error located in field 15:5 is valid.</td>
<td></td>
</tr>
</tbody>
</table>
# LBCF DPF Error log register 10

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0B158h</td>
</tr>
</tbody>
</table>

Slice 0 Bank 2 subbank2 Error log register

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:21</td>
<td><strong>Row Number for Error 1</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Row Number for Error1: The physical row Addr where the parity error has been detected in SLM Bank. The number of rows varies between 4K vs. 8K/16K subbanks which requires 10 bits vs. 11 bits respectively. This field contains the row# with the error. Slice 0 Bank 2 Error log register 00.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>20:17</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td></td>
<td><strong>Valid Error 1</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Error: The error located in field 31:21 is valid. Slice 0 Bank 2 Error log register 00.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>15:5</td>
<td></td>
<td><strong>Row Number for Error0</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Row Number for Error0: The physical row Addr where the parity error has been detected in SLM Bank. Slice 0 Bank 2 Error log register 00.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4:1</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td><strong>Valid Error 0</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Error: The error located in field 15:5 is valid. Slice 0 Bank 2 Error log register 00.</td>
</tr>
</tbody>
</table>
LBCF DPF Error log register 11

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:21</td>
<td><strong>Row Number for Error 1</strong>&lt;br&gt;&lt;ul&gt;&lt;li&gt;Access: R/W&lt;/li&gt;&lt;li&gt;Row Number for Error 1: The physical row Addr where the parity error has been detected in SLM Bank.&lt;br&gt;The number of rows varies between 4K vs. 8K/16K subbanks which requires 10 bits vs. 11 bits respectively.&lt;br&gt;This field contains the row# with the error.&lt;br&gt;Slice 0 Bank 2 Error log register 00.&lt;/li&gt;&lt;/ul&gt;</td>
</tr>
<tr>
<td></td>
<td>20:17</td>
<td><strong>Reserved</strong>&lt;br&gt;&lt;ul&gt;&lt;li&gt;Access: RO&lt;/li&gt;&lt;/ul&gt;</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td><strong>Valid Error 1</strong>&lt;br&gt;&lt;ul&gt;&lt;li&gt;Access: R/W&lt;/li&gt;&lt;li&gt;Valid Error: The error located in field 31:21 is valid.&lt;br&gt;Slice 0 Bank 2 Error log register 00.&lt;/li&gt;&lt;/ul&gt;</td>
</tr>
<tr>
<td>15:5</td>
<td></td>
<td><strong>Row Number for Error 0</strong>&lt;br&gt;&lt;ul&gt;&lt;li&gt;Access: R/W&lt;/li&gt;&lt;li&gt;Row Number for Error 0: The physical row Addr where the parity error has been detected in SLM Bank.&lt;br&gt;Slice 0 Bank 2 Error log register 00.&lt;/li&gt;&lt;/ul&gt;</td>
</tr>
<tr>
<td></td>
<td>4:1</td>
<td><strong>Reserved</strong>&lt;br&gt;&lt;ul&gt;&lt;li&gt;Access: RO&lt;/li&gt;&lt;/ul&gt;</td>
</tr>
</tbody>
</table>
| 0     |      | **Valid Error 0**<br><ul><li>Access: R/W</li><li>Valid Error: The error located in field 15:5 is valid.<br>Slice 0 Bank 2 Error log register 00.</li></ul>
### LBCF DPF Error log register 12

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:21</td>
<td><strong>Row Number for Error 1</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Row Number for Error 1: The physical row Addr where the parity error has been detected in SLM Bank. Slice 0 Bank 3 Error log register 00.</td>
<td></td>
</tr>
<tr>
<td>20:17</td>
<td></td>
<td><strong>Reserved</strong></td>
<td>RO</td>
</tr>
<tr>
<td>16</td>
<td>15:5</td>
<td><strong>Row Number for Error 0</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Row Number for Error 0: The physical row Addr where the parity error has been detected in SLM Bank. Slice 0 Bank 3 Error log register 00.</td>
<td></td>
</tr>
<tr>
<td>4:1</td>
<td>0</td>
<td><strong>Reserved</strong></td>
<td>RO</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td><strong>Valid Error 0</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Error: The error located in field 15:5 is valid. Slice 0 Bank 3 Error log register 00.</td>
<td></td>
</tr>
</tbody>
</table>
### LBCFERRLOG10 - LBCF DPF Error log register 13

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0B164h</td>
</tr>
</tbody>
</table>

#### Slice 0 Bank 3 subbank1 Error log register 00

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:21</td>
<td><strong>Row Number for Error 1</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Row Number for Error 1:</strong> The physical row Addr where the parity error has been detected in SLM Bank. Slice 0 Bank 3 Error log register 00.</td>
<td></td>
</tr>
<tr>
<td>20:17</td>
<td>Reserved</td>
<td>Access: RO</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
<td><strong>Valid Error 1</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Valid Error:</strong> The error located in field 31:21 is valid. Slice 0 Bank 3 Error log register 00.</td>
<td></td>
</tr>
<tr>
<td>15:5</td>
<td><strong>Row Number for Error 0</strong></td>
<td>Access: R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Row Number for Error 0:</strong> The physical row Addr where the parity error has been detected in SLM Bank. Slice 0 Bank 3 Error log register 00.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4:1</td>
<td><strong>Reserved</strong></td>
<td>Access: RO</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td><strong>Valid Error 0</strong></td>
<td>Access: R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Valid Error:</strong> The error located in field 15:5 is valid.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## LBCF DPF Error log register 14

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:21</td>
<td><strong>Row Number for Error 1</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Row Number for Error1: The physical row Addr where the parity error has been detected in SLM Bank. Slice 0 Bank 3 Error log register 00.</td>
</tr>
<tr>
<td>20:17</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td><strong>Valid Error 1</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Error: The error located in field 31:21 is valid. Slice 0 Bank 3 Error log register 00.</td>
</tr>
<tr>
<td>15:5</td>
<td></td>
<td><strong>Row Number for Error0</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Row Number for Error0: The physical row Addr where the parity error has been detected in SLM Bank. Slice 0 Bank 3 Error log register 00.</td>
</tr>
<tr>
<td>4:1</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td><strong>Valid Error 0</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Error: The error located in field 15:5 is valid. Slice 0 Bank 3 Error log register 00.</td>
</tr>
</tbody>
</table>
## LBCF DPF Error log register 15

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0B16Ch</td>
</tr>
</tbody>
</table>

### Description

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:21</td>
<td><strong>Row Number for Error 1</strong>&lt;br&gt;Access: R/W&lt;br&gt;Row Number for Error 1: The physical row Addr where the parity error has been detected in SLM Bank. Slice 0 Bank 3 Error log register 00.</td>
</tr>
<tr>
<td>20:17</td>
<td></td>
<td><strong>Reserved</strong>&lt;br&gt;Access: RO&lt;br&gt;Valid Error 1 has no corresponding Row Number for Error 1.</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td><strong>Valid Error 1</strong>&lt;br&gt;Access: R/W&lt;br&gt;Valid Error: The error located in field 31:21 is valid. Slice 0 Bank 3 Error log register 00.</td>
</tr>
<tr>
<td>15:5</td>
<td></td>
<td><strong>Row Number for Error0</strong>&lt;br&gt;Access: R/W&lt;br&gt;Row Number for Error 0: The physical row Addr where the parity error has been detected in SLM Bank. Slice 0 Bank 3 Error log register 00.</td>
</tr>
<tr>
<td>4:1</td>
<td></td>
<td><strong>Reserved</strong>&lt;br&gt;Access: RO&lt;br&gt;Valid Error 0 has no corresponding Row Number for Error 0.</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td><strong>Valid Error 0</strong>&lt;br&gt;Access: R/W&lt;br&gt;Valid Error: The error located in field 15:5 is valid. Slice 0 Bank 3 Error log register 00.</td>
</tr>
</tbody>
</table>
LBCF DPF Error log register 16

<table>
<thead>
<tr>
<th>LBCFERRLOG13 - LBCF DPF Error log register 16</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 0B170h</td>
</tr>
</tbody>
</table>

Slice 1 bank 0 Subbank0 Error log register

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:21</td>
<td><strong>Row Number for Error 1</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Row Number for Error1: The physical row Addr where the parity error has been detected in SLM Bank. Slice 1 bank 1 Error log register 00.</td>
<td></td>
</tr>
<tr>
<td>20:17</td>
<td></td>
<td><strong>Reserved</strong></td>
<td>RO</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td><strong>Valid Error 1</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Error: The error located in field 31:21 is valid. Slice 1 bank 1 Error log register 00.</td>
<td></td>
</tr>
<tr>
<td>15:5</td>
<td></td>
<td><strong>Row Number for Error0</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Row Number for Error0: The physical row Addr where the parity error has been detected in SLM Bank. Slice 1 bank 1 Error log register 00.</td>
<td></td>
</tr>
<tr>
<td>4:1</td>
<td></td>
<td><strong>Reserved</strong></td>
<td>RO</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td><strong>Valid Error 0</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Error: The error located in field 15:5 is valid. Slice 1 bank 1 Error log register 00.</td>
<td></td>
</tr>
</tbody>
</table>
## LBCF DPF Error log register 17

### LBCFERRLOG14 - LBCF DPF Error log register 17

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0B174h</td>
</tr>
</tbody>
</table>

Slice 1 bank 0 subbank1 Error log register

<table>
<thead>
<tr>
<th>DWORD</th>
<th>BIT</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:21</td>
<td><strong>Row Number for Error 1</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Row Number for Error 1</strong>: The physical row Addr where the parity error has been detected in SLM Bank. Slice 1 bank 1 Error log register 00.</td>
<td></td>
</tr>
<tr>
<td>20:17</td>
<td></td>
<td><strong>Reserved</strong></td>
<td>RO</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td><strong>Valid Error 1</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Valid Error</strong>: The error located in field 31:21 is valid. Slice 1 bank 1 Error log register 00.</td>
<td></td>
</tr>
<tr>
<td>15:5</td>
<td></td>
<td><strong>Row Number for Error 0</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Row Number for Error 0</strong>: The physical row Addr where the parity error has been detected in SLM Bank. Slice 1 bank 1 Error log register 00.</td>
<td></td>
</tr>
<tr>
<td>4:1</td>
<td></td>
<td><strong>Reserved</strong></td>
<td>RO</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td><strong>Valid Error 0</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Valid Error</strong>: The error located in field 15:5 is valid.</td>
<td></td>
</tr>
</tbody>
</table>
### LBCFERRLOG15 - LBCF DPF Error log register 18

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:21</td>
<td><strong>Row Number for Error 1</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Row Number for Error1: The physical row Addr where the parity error has been detected in SLM Bank.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Slice 1 bank 1 Error log register 00.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>20:17</td>
<td><strong>Reserved</strong></td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td><strong>Valid Error 1</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Error: The error located in field 31:21 is valid.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Slice 1 bank 1 Error log register 00.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>15:5</td>
<td><strong>Row Number for Error0</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Row Number for Error0: The physical row Addr where the parity error has been detected in SLM Bank.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Slice 1 bank 1 Error log register 00.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4:1</td>
<td><strong>Reserved</strong></td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td><strong>Valid Error 0</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Error: The error located in field 15:5 is valid.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Slice 1 bank 1 Error log register 00.</td>
<td></td>
</tr>
</tbody>
</table>
### LBCF DPF Error log register 19

<table>
<thead>
<tr>
<th>LBCFERRLOG16 - LBCF DPF Error log register 19</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 0B17Ch</td>
</tr>
<tr>
<td>Slice1 bank 0 subbank3 Error log register</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:21</td>
<td><strong>Row Number for Error 1</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Row Number for Error 1: The physical row Addr where the parity error has been detected in SLM Bank. Slice 1 bank 1 Error log register 00.</td>
</tr>
<tr>
<td>20:17</td>
<td>Reserved</td>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td><strong>Valid Error 1</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Error: The error located in field 31:21 is valid. Slice 1 bank 1 Error log register 00.</td>
</tr>
<tr>
<td>15:5</td>
<td></td>
<td><strong>Row Number for Error 0</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Row Number for Error 0: The physical row Addr where the parity error has been detected in SLM Bank. Slice 1 bank 1 Error log register 00.</td>
</tr>
<tr>
<td>4:1</td>
<td>Reserved</td>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td><strong>Valid Error 0</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Error: The error located in field 15:5 is valid. Slice 1 bank 1 Error log register 00.</td>
</tr>
</tbody>
</table>
## LBCF DPF Error log register 20

### LBCFERRLOG17 - LBCF DPF Error log register 20

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:21</td>
<td><strong>Row Number for Error 1</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Row Number for Error 1: The physical row Addr where the parity error has been detected in SLM Bank. Slice 1 bank 0 Error log register 00.</td>
</tr>
<tr>
<td>20:17</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td><strong>Valid Error 1</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Error: The error located in field 31:21 is valid. Slice 1 bank 0 Error log register 00.</td>
</tr>
<tr>
<td>15:5</td>
<td></td>
<td><strong>Row Number for Error 0</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Row Number for Error 0: The physical row Addr where the parity error has been detected in SLM Bank. Slice 1 bank 0 Error log register 00.</td>
</tr>
<tr>
<td>4:1</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td><strong>Valid Error 0</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Error: The error located in field 15:5 is valid. Slice 1 bank 0 Error log register 00.</td>
</tr>
</tbody>
</table>
## LBCF DPF Error log register 21

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:21</td>
<td>Row Number for Error 1</td>
<td>R/W</td>
<td>Row Number for Error1: The physical row Addr where the parity error has been detected in SLM Bank. Slice 1 bank 0 Error log register 00.</td>
</tr>
<tr>
<td></td>
<td>20:17</td>
<td>Reserved</td>
<td>RO</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>15:5</td>
<td>Row Number for Error0</td>
<td>R/W</td>
<td>Row Number for Error0: The physical row Addr where the parity error has been detected in SLM Bank. Slice 1 bank 0 Error log register 00.</td>
</tr>
<tr>
<td></td>
<td>4:1</td>
<td>Reserved</td>
<td>RO</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>Valid Error 0</td>
<td>R/W</td>
<td>Valid Error: The error located in field 15:5 is valid.</td>
</tr>
</tbody>
</table>
# LBCF DPF Error log register 22

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:21</td>
<td><strong>Row Number for Error 1</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Row Number for Error1: The physical row Addr where the parity error has been detected in SLM Bank. Slice 1 bank 0 Error log register 00.</td>
<td></td>
</tr>
<tr>
<td>20:17</td>
<td></td>
<td><strong>Reserved</strong></td>
<td>RO</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td><strong>Valid Error 1</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Error: The error located in field 31:21 is valid. Slice 1 bank 0 Error log register 00.</td>
<td></td>
</tr>
<tr>
<td>15:5</td>
<td></td>
<td><strong>Row Number for Error0</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Row Number for Error0: The physical row Addr where the parity error has been detected in SLM Bank. Slice 1 bank 0 Error log register 00.</td>
<td></td>
</tr>
<tr>
<td>4:1</td>
<td></td>
<td><strong>Reserved</strong></td>
<td>RO</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td><strong>Valid Error 0</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Error: The error located in field 15:5 is valid. Slice 1 bank 0 Error log register 00.</td>
<td></td>
</tr>
</tbody>
</table>
# LBCF DPF Error log register 23

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:21</td>
<td><strong>Row Number for Error 1</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Row Number for Error 1: The physical row Addr where the parity error has been detected in SLM Bank.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Slice 1 bank 0 Error log register 00.</td>
<td></td>
</tr>
<tr>
<td>20:17</td>
<td></td>
<td><strong>Reserved</strong></td>
<td>RO</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td><strong>Valid Error 1</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Error: The error located in field 31:21 is valid.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Slice 1 bank 0 Error log register 00.</td>
<td></td>
</tr>
<tr>
<td>15:5</td>
<td></td>
<td><strong>Row Number for Error 0</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Row Number for Error 0: The physical row Addr where the parity error has been detected in SLM Bank.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Slice 1 bank 0 Error log register 00.</td>
<td></td>
</tr>
<tr>
<td>4:1</td>
<td></td>
<td><strong>Reserved</strong></td>
<td>RO</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td><strong>Valid Error 0</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Error: The error located in field 15:5 is valid.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Slice 1 bank 0 Error log register 00.</td>
<td></td>
</tr>
</tbody>
</table>
### LBCFERRLOG21 - LBCF DPF Error log register 24

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0B190h</td>
</tr>
</tbody>
</table>

Slice 1 bank 2 subbank0 Error log register 00

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:21</td>
<td><strong>Row Number for Error 1</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Row Number for Error 1: The physical row Addr where the parity error has been detected in SLM Bank. Slice 1 bank 2 Error log register 00.</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>20:17</td>
<td><strong>Reserved</strong></td>
<td>RO</td>
</tr>
<tr>
<td>0</td>
<td>16</td>
<td><strong>Valid Error 1</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Error: The error located in field 31:21 is valid. Slice 1 bank 2 Error log register 00.</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>15:5</td>
<td><strong>Row Number for Error 0</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Row Number for Error 0: The physical row Addr where the parity error has been detected in SLM Bank. Slice 1 bank 2 Error log register 00.</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>4:1</td>
<td><strong>Reserved</strong></td>
<td>RO</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td><strong>Valid Error 0</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Error: The error located in field 15:5 is valid. Slice 1 bank 2 Error log register 00.</td>
<td></td>
</tr>
</tbody>
</table>
### LBCF DPF Error log register 25

**LBCFERRLOG22** - LBCF DPF Error log register 25

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0B194h</td>
</tr>
</tbody>
</table>

Slice 1 bank 2 subbank 1 Error log register 00

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:21</td>
<td><strong>Row Number for Error 1</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Row Number for Error1: The physical row Addr where the parity error has been detected in SLM Bank. Slice 1 bank 2 Error log register 00.</td>
<td></td>
</tr>
<tr>
<td>20:17</td>
<td></td>
<td><strong>Reserved</strong></td>
<td>RO</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td><strong>Valid Error 1</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Error: The error located in field 31:21 is valid. Slice 1 bank 2 Error log register 00.</td>
<td></td>
</tr>
<tr>
<td>15:5</td>
<td></td>
<td><strong>Row Number for Error0</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Row Number for Error0: The physical row Addr where the parity error has been detected in SLM Bank. Slice 1 bank 2 Error log register 00.</td>
<td></td>
</tr>
<tr>
<td>4:1</td>
<td></td>
<td><strong>Reserved</strong></td>
<td>RO</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td><strong>Valid Error 0</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Error: The error located in field 15:5 is valid.</td>
<td></td>
</tr>
</tbody>
</table>
## LBCF DPF Error log register 26

<table>
<thead>
<tr>
<th>LBCFERRLOG23 - LBCF DPF Error log register 26</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 0B198h</td>
</tr>
</tbody>
</table>

Slice 1 bank 2 subbank 2 Error log register

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:21</td>
<td><strong>Row Number for Error 1</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Row Number for Error 1:</strong> The physical row Addr where the parity error has been detected in SLM Bank. Slice 1 bank 2 Error log register 00.</td>
</tr>
<tr>
<td>20:17</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td><strong>Valid Error 1</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Valid Error:</strong> The error located in field 31:21 is valid. Slice 1 bank 2 Error log register 00.</td>
</tr>
<tr>
<td>15:5</td>
<td></td>
<td><strong>Row Number for Error 0</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Row Number for Error 0:</strong> The physical row Addr where the parity error has been detected in SLM Bank. Slice 1 bank 2 Error log register 00.</td>
</tr>
<tr>
<td>4:1</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td><strong>Valid Error 0</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Valid Error:</strong> The error located in field 15:5 is valid. Slice 1 bank 2 Error log register 00.</td>
</tr>
</tbody>
</table>
## LBCF DPF Error log register 27

### LBCFERRLOG24 - LBCF DPF Error log register 27

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0B19Ch</td>
</tr>
</tbody>
</table>

Slice 1 bank 2 subbank 3 Error log register

### Description

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:21</td>
<td><strong>Row Number for Error 1</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Row Number for Error1: The physical row Addr where the parity error has been detected in SLM Bank. Slice 1 bank 2 Error log register 00.</td>
</tr>
<tr>
<td>0</td>
<td>20:17</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td>0</td>
<td>16</td>
<td><strong>Valid Error 1</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Error: The error located in field 31:21 is valid. Slice 1 bank 2 Error log register 00.</td>
</tr>
<tr>
<td>0</td>
<td>15:5</td>
<td><strong>Row Number for Error0</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Row Number for Error0: The physical row Addr where the parity error has been detected in SLM Bank. Slice 1 bank 2 Error log register 00.</td>
</tr>
<tr>
<td>0</td>
<td>4:1</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td><strong>Valid Error 0</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Error: The error located in field 15:5 is valid. Slice 1 bank 2 Error log register 00.</td>
</tr>
</tbody>
</table>
### LBCFERRLOG25 - LBCF DPF Error log register 28

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0B1A0h</td>
</tr>
</tbody>
</table>

#### Description

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:21</td>
<td><strong>Row Number for Error 1</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Row Number for Error 1: The physical row Addr where the parity error has been detected in SLM Bank. Slice 1 Bank 3 Error log register 00.</td>
</tr>
<tr>
<td>20:17</td>
<td>Reserved</td>
<td>Access: RO</td>
</tr>
<tr>
<td>16</td>
<td>15:5</td>
<td><strong>Row Number for Error 0</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Row Number for Error 0: The physical row Addr where the parity error has been detected in SLM Bank. Slice 1 Bank 3 Error log register 00.</td>
</tr>
<tr>
<td>4:1</td>
<td>Reserved</td>
<td>Access: RO</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td><strong>Valid Error 0</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Error: The error located in field 15:5 is valid. Slice 1 Bank 3 Error log register 00.</td>
</tr>
</tbody>
</table>
### LBCF DPF Error log register 29

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:21</td>
<td><strong>Row Number for Error 1</strong>&lt;br&gt;Access: R/W&lt;br&gt;Row Number for Error1: The physical row Addr where the parity error has been detected in SLM Bank.&lt;br&gt;Slice 1 Bank 3 Error log register 00.</td>
</tr>
<tr>
<td>20:17</td>
<td></td>
<td><strong>Reserved</strong>&lt;br&gt;Access: RO</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td><strong>Valid Error 1</strong>&lt;br&gt;Access: R/W&lt;br&gt;Valid Error: The error located in field 31:21 is valid.&lt;br&gt;Slice 1 Bank 3 Error log register 00.</td>
</tr>
<tr>
<td>15:5</td>
<td></td>
<td><strong>Row Number for Error0</strong>&lt;br&gt;Access: R/W&lt;br&gt;Row Number for Error0: The physical row Addr where the parity error has been detected in SLM Bank.&lt;br&gt;Slice 1 Bank 3 Error log register 00.</td>
</tr>
<tr>
<td>4:1</td>
<td></td>
<td><strong>Reserved</strong>&lt;br&gt;Access: RO</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td><strong>Valid Error 0</strong>&lt;br&gt;Access: R/W&lt;br&gt;Valid Error: The error located in field 15:5 is valid.</td>
</tr>
</tbody>
</table>
# LBCF DPF Error log register 30

## LBCFERRLOG27 - LBCF DPF Error log register 30

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:21</td>
<td><strong>Row Number for Error 1</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Row Number for Error 1:</strong> The physical row Addr where the parity error has been detected in SLM Bank.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Slice 1 Bank 3 Error log register 00.</td>
<td></td>
</tr>
<tr>
<td>20:17</td>
<td></td>
<td><strong>Reserved</strong></td>
<td>RO</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td><strong>Valid Error 1</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Valid Error: The error located in field 31:21 is valid.</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Slice 1 Bank 3 Error log register 00.</td>
<td></td>
</tr>
<tr>
<td>15:5</td>
<td></td>
<td><strong>Row Number for Error 0</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Row Number for Error 0:</strong> The physical row Addr where the parity error has been detected in SLM Bank.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Slice 1 Bank 3 Error log register 00.</td>
<td></td>
</tr>
<tr>
<td>4:1</td>
<td></td>
<td><strong>Reserved</strong></td>
<td>RO</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td><strong>Valid Error 0</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Valid Error: The error located in field 15:5 is valid.</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Slice 1 Bank 3 Error log register 00.</td>
<td></td>
</tr>
</tbody>
</table>
## LBCF DPF Error log register 31

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:21</td>
<td><strong>Row Number for Error 1</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Row Number for Error1: The physical row Addr where the parity error has been detected in SLM Bank. Slice 1 Bank 3 Error log register 00.</td>
</tr>
<tr>
<td>20:17</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td>16</td>
<td>15:5</td>
<td><strong>Row Number for Error0</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Row Number for Error0: The physical row Addr where the parity error has been detected in SLM Bank. Slice 1 Bank 3 Error log register 00.</td>
</tr>
<tr>
<td>4:1</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td><strong>Valid Error 0</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Error: The error located in field 15:5 is valid. Slice 1 Bank 3 Error log register 00.</td>
</tr>
</tbody>
</table>
# LBCF DPF Error log register 32

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:21</td>
<td><strong>Row Number for Error 1</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Row Number for Error1: The physical row Addr where the parity error has been detected in SLM Bank. Slice 2 Bank 0 Error log register 00.</td>
<td></td>
</tr>
<tr>
<td>20:17</td>
<td></td>
<td><strong>Reserved</strong></td>
<td>RO</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td><strong>Valid Error 1</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Error: The error located in field 31:21 is valid. Slice 2 Bank 0 Error log register 00.</td>
<td></td>
</tr>
<tr>
<td>15:5</td>
<td></td>
<td><strong>Row Number for Error0</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Row Number for Error0: The physical row Addr where the parity error has been detected in SLM Bank. Slice 2 Bank 0 Error log register 00.</td>
<td></td>
</tr>
<tr>
<td>4:1</td>
<td></td>
<td><strong>Reserved</strong></td>
<td>RO</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td><strong>Valid Error 0</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Error: The error located in field 15:5 is valid. Slice 2 Bank 0 Error log register 00.</td>
<td></td>
</tr>
</tbody>
</table>
### LBCF DPF Error log register 33

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:21</td>
<td><strong>Row Number for Error 1</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Row Number for Error1: The physical row Addr where the parity error has been detected in SLM Bank. Slice 2 Bank 0 Error log register 00.</td>
</tr>
<tr>
<td>20:17</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td>16</td>
<td></td>
<td><strong>Valid Error 1</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Error: The error located in field 31:21 is valid. Slice 2 Bank 0 Error log register 00.</td>
</tr>
<tr>
<td>15:5</td>
<td></td>
<td><strong>Row Number for Error0</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Row Number for Error0: The physical row Addr where the parity error has been detected in SLM Bank. Slice 2 Bank 0 Error log register 00.</td>
</tr>
<tr>
<td>4:1</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td><strong>Valid Error 0</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
</tbody>
</table>
|       |          | Valid Error: The error located in field 15:5 is valid. 
### LBCFERRLOG31 - LBCF DPF Error log register 34

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
</table>
| 0     | 31:21 | **Row Number for Error 1**  
|       |       | **Access:** R/W  
|       |       | Row Number for Error 1: The physical row Addr where the parity error has been detected in SLM Bank.  
|       |       | Slice 2 Bank 0 Error log register 00. |
| 20:17 |      | **Reserved**  
|       |       | **Access:** RO  |
| 16    |      | **Valid Error 1**  
|       |       | **Access:** R/W  
|       |       | Valid Error: The error located in field 31:21 is valid.  
|       |       | Slice 2 Bank 0 Error log register 00. |
| 15:5  |      | **Row Number for Error 0**  
|       |       | **Access:** R/W  
|       |       | Row Number for Error 0: The physical row Addr where the parity error has been detected in SLM Bank.  
|       |       | Slice 2 Bank 0 Error log register 00. |
| 4:1   |      | **Reserved**  
|       |       | **Access:** RO  |
| 0     |      | **Valid Error 0**  
|       |       | **Access:** R/W  
|       |       | Valid Error: The error located in field 15:5 is valid.  
|       |       | Slice 2 Bank 0 Error log register 00. |
LBCF DPF Error log register 35

<table>
<thead>
<tr>
<th>LBCFERRLOG32 - LBCF DPF Error log register 35</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 0B1BCh</td>
</tr>
</tbody>
</table>

Slice 2 Bank 0 Subbank 3 Error log register

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:21</td>
<td>Row Number for Error 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Row Number for Error 1: The physical row Addr where the parity error has been detected in SLM Bank. Slice 2 Bank 0 Error log register 00.</td>
</tr>
<tr>
<td>20:17</td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td>Valid Error 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Error: The error located in field 31:21 is valid. Slice 2 Bank 0 Error log register 00.</td>
</tr>
<tr>
<td>15:5</td>
<td></td>
<td>Row Number for Error 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Row Number for Error 0: The physical row Addr where the parity error has been detected in SLM Bank. Slice 2 Bank 0 Error log register 00.</td>
</tr>
<tr>
<td>4:1</td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td>Valid Error 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Error: The error located in field 15:5 is valid. Slice 2 Bank 0 Error log register 00.</td>
</tr>
</tbody>
</table>
**LBCF DPF Error log register 36**

<table>
<thead>
<tr>
<th><strong>LBCFERRLOG33 - LBCF DPF Error log register 36</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 0B1C0h</td>
</tr>
</tbody>
</table>

**Slice 2 Bank 1 Subbank 0 Error log register**

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:21</td>
<td><strong>Row Number for Error 1</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Row Number for Error 1: The physical row Addr where the parity error has been detected in SLM Bank. Slice 2 Bank 1 Error log register 00.</td>
</tr>
<tr>
<td>20:17</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td><strong>Valid Error 1</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Error: The error located in field 31:21 is valid. Slice 2 Bank 1 Error log register 00.</td>
</tr>
<tr>
<td>15:5</td>
<td></td>
<td><strong>Row Number for Error0</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Row Number for Error0: The physical row Addr where the parity error has been detected in SLM Bank. Slice 2 Bank 1 Error log register 00.</td>
</tr>
<tr>
<td>4:1</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td><strong>Valid Error 0</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Error: The error located in field 15:5 is valid. Slice 2 Bank 1 Error log register 00.</td>
</tr>
</tbody>
</table>
## LBCF DPF Error log register 37

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:21</td>
<td><strong>Row Number for Error 1</strong></td>
<td>R/W</td>
<td>Row Number for Error1: The physical row Addr where the parity error has been detected in SLM Bank. Slice 2 Bank 1 Error log register 00.</td>
</tr>
<tr>
<td></td>
<td>20:17</td>
<td><strong>Reserved</strong></td>
<td>RO</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>15:5</td>
<td><strong>Row Number for Error0</strong></td>
<td>R/W</td>
<td>Row Number for Error0: The physical row Addr where the parity error has been detected in SLM Bank. Slice 2 Bank 1 Error log register 00.</td>
</tr>
<tr>
<td>4:1</td>
<td>0</td>
<td><strong>Reserved</strong></td>
<td>RO</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td><strong>Valid Error 0</strong></td>
<td>R/W</td>
<td>Valid Error: The error located in field 15:5 is valid.</td>
</tr>
</tbody>
</table>
## LBCF DPF Error log register 38

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0B1C8h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
</table>
| 0     | 31:21 | **Row Number for Error 1**  
|       |       | Access: R/W  
|       |       | Row Number for Error1: The physical row Addr where the parity error has been detected in SLM Bank. Slice 2 Bank 1 Error log register 00. |
| 20:17 | Reserved | Access: RO  
| 16    | **Valid Error 1**  
|       | Access: R/W  
|       | Valid Error: The error located in field 31:21 is valid. Slice 2 Bank 1 Error log register 00. |
| 15:5  | **Row Number for Error0**  
|       | Access: R/W  
|       | Row Number for Error0: The physical row Addr where the parity error has been detected in SLM Bank. Slice 2 Bank 1 Error log register 00. |
| 4:1   | Reserved | Access: RO  
| 0     | **Valid Error 0**  
|       | Access: R/W  
|       | Valid Error: The error located in field 15:5 is valid. Slice 2 Bank 1 Error log register 00. |
# LBCF DPF Error log register 39

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
</table>
| 0     | 31:21 | **Row Number for Error 1**  
|       |       | Access: R/W  
|       |       | Row Number for Error1: The physical row Addr where the parity error has been detected in SLM Bank.  
|       |       | Slice 2 Bank 1 Error log register 00. |
| 20:17 |       | **Reserved**  
|       |       | Access: RO  |
| 16    |       | **Valid Error 1**  
|       |       | Access: R/W  
|       |       | Valid Error: The error located in field 31:21 is valid.  
|       |       | Slice 2 Bank 1 Error log register 00. |
| 15:5  |       | **Row Number for Error0**  
|       |       | Access: R/W  
|       |       | Row Number for Error0: The physical row Addr where the parity error has been detected in SLM Bank.  
|       |       | Slice 2 Bank 1 Error log register 00. |
| 4:1   |       | **Reserved**  
|       |       | Access: RO  |
| 0     |       | **Valid Error 0**  
|       |       | Access: R/W  
|       |       | Valid Error: The error located in field 15:5 is valid.  
|       |       | Slice 2 Bank 1 Error log register 00. |
### LBCFERRLOG37 - LBCF DPF Error log register 40

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td><strong>Row Number for Error 1</strong></td>
</tr>
<tr>
<td></td>
<td>31:21</td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Row Number for Error 1: The physical row Addr where the parity error has been detected in SLM Bank.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Slice 2 Bank 2 Error log register 00.</td>
</tr>
<tr>
<td>20:17</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td><strong>Valid Error 1</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Error: The error located in field 31:21 is valid.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Slice 2 Bank 2 Error log register 00.</td>
</tr>
<tr>
<td>15:5</td>
<td></td>
<td><strong>Row Number for Error 0</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Row Number for Error 0: The physical row Addr where the parity error has been detected in SLM Bank.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Slice 2 Bank 2 Error log register 00.</td>
</tr>
<tr>
<td>4:1</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td><strong>Valid Error 0</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Error: The error located in field 15:5 is valid.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Slice 2 Bank 2 Error log register 00.</td>
</tr>
</tbody>
</table>
## LBCF DPF Error log register 41

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
</table>
| 0     | 31:21 | **Row Number for Error 1**  
|       |       | Access: R/W |  
|       |       | Row Number for Error 1: The physical row Addr where the parity error has been detected in SLM Bank.  
|       |       | Slice 2 Bank 2 Error log register 00. |
| 20:17 |       | **Reserved**  
|       |       | Access: RO |  
| 16    |       | **Valid Error 1**  
|       |       | Access: R/W |  
|       |       | Valid Error: The error located in field 31:21 is valid.  
|       |       | Slice 2 Bank 2 Error log register 00. |
| 15:5  |       | **Row Number for Error 0**  
|       |       | Access: R/W |  
|       |       | Row Number for Error 0: The physical row Addr where the parity error has been detected in SLM Bank.  
|       |       | Slice 2 Bank 2 Error log register 00. |
| 4:1   |       | **Reserved**  
|       |       | Access: RO |  
| 0     |       | **Valid Error 0**  
|       |       | Access: R/W |  
|       |       | Valid Error: The error located in field 15:5 is valid. |
## LBCF DPF Error log register 42

### LBCFERRLOG39 - LBCF DPF Error log register 42

<table>
<thead>
<tr>
<th>Register Space</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits)</td>
<td>32</td>
</tr>
<tr>
<td>Address</td>
<td>0B1D8h</td>
</tr>
</tbody>
</table>

**Slice 2 Bank 2 subbank 2 Error log register**

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:21</td>
<td><strong>Row Number for Error 1</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Row Number for Error 1: The physical row Addr where the parity error has been detected in SLM Bank. Slice 2 Bank 2 Error log register 00.</td>
</tr>
<tr>
<td>20:17</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td><strong>Valid Error 1</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Error: The error located in field 31:21 is valid. Slice 2 Bank 2 Error log register 00.</td>
</tr>
<tr>
<td>15:5</td>
<td></td>
<td><strong>Row Number for Error 0</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Row Number for Error 0: The physical row Addr where the parity error has been detected in SLM Bank. Slice 2 Bank 2 Error log register 00.</td>
</tr>
<tr>
<td>4:1</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td><strong>Valid Error 0</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Error: The error located in field 15:5 is valid. Slice 2 Bank 2 Error log register 00.</td>
</tr>
</tbody>
</table>
# LBCF DPF Error log register 43

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0B1DCh</td>
</tr>
</tbody>
</table>

Slice 2 Bank 2 subbank 3 Error log register

<table>
<thead>
<tr>
<th>DWORD</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:21</td>
<td><strong>Row Number for Error 1</strong>&lt;br&gt;Access: R/W&lt;br&gt;Row Number for Error 1: The physical row Addr where the parity error has been detected in SLM Bank. Slice 2 Bank 2 Error log register 00.</td>
</tr>
<tr>
<td>0</td>
<td>20:17</td>
<td><strong>Reserved</strong>&lt;br&gt;Access: RO&lt;br&gt;Row Number for Error 0: The physical row Addr where the parity error has been detected in SLM Bank. Slice 2 Bank 2 Error log register 00.</td>
</tr>
<tr>
<td>0</td>
<td>16</td>
<td><strong>Valid Error 1</strong>&lt;br&gt;Access: R/W&lt;br&gt;Valid Error: The error located in field 31:21 is valid. Slice 2 Bank 2 Error log register 00.</td>
</tr>
<tr>
<td>0</td>
<td>15:5</td>
<td><strong>Row Number for Error 0</strong>&lt;br&gt;Access: R/W&lt;br&gt;Row Number for Error 0: The physical row Addr where the parity error has been detected in SLM Bank. Slice 2 Bank 2 Error log register 00.</td>
</tr>
<tr>
<td>0</td>
<td>4:1</td>
<td><strong>Reserved</strong>&lt;br&gt;Access: RO&lt;br&gt;Row Number for Error 1: The physical row Addr where the parity error has been detected in SLM Bank. Slice 2 Bank 2 Error log register 00.</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td><strong>Valid Error 0</strong>&lt;br&gt;Access: R/W&lt;br&gt;Valid Error: The error located in field 15:5 is valid. Slice 2 Bank 2 Error log register 00.</td>
</tr>
</tbody>
</table>
### LBCFERRLOG41 - LBCF DPF Error log register 44

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0B1E0h</td>
</tr>
</tbody>
</table>

Slice 2 Bank 3 subbank 0 Error log register

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:21</td>
<td><strong>Row Number for Error 1</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Row Number for Error1: The physical row Addr where the parity error has been detected in SLM Bank. Slice 2 Bank 3 Error log register 00.</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>20:17</td>
<td><strong>Reserved</strong></td>
<td>RO</td>
</tr>
<tr>
<td>0</td>
<td>16</td>
<td><strong>Valid Error 1</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Error: The error located in field 31:21 is valid. Slice 2 Bank 3 Error log register 00.</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>15:5</td>
<td><strong>Row Number for Error0</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Row Number for Error0: The physical row Addr where the parity error has been detected in SLM Bank. and Slice 2 Bank 3 Error log register 00.</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>4:1</td>
<td><strong>Reserved</strong></td>
<td>RO</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td><strong>Valid Error 0</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Error: The error located in field 15:5 is valid corresponding logical 16KB group should bypass this row. Slice 2 Bank 3 Error log register 00.</td>
<td></td>
</tr>
</tbody>
</table>
## LBCF DPF Error log register 45

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:21</td>
<td><strong>Row Number for Error 1</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>[R/W]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Row Number for Error 1: The physical row Addr where the parity error has been detected in SLM Bank. Slice 2 Bank 3 Error log register 00.</td>
</tr>
<tr>
<td>20:17</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>[RO]</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td><strong>Valid Error 1</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>[R/W]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Error: The error located in field 31:21 is valid. Slice 2 Bank 3 Error log register 00.</td>
</tr>
<tr>
<td>15:5</td>
<td></td>
<td><strong>Row Number for Error 0</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>[R/W]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Row Number for Error 0: The physical row Addr where the parity error has been detected in SLM Bank. Slice 2 Bank 3 Error log register 00.</td>
</tr>
<tr>
<td>4:1</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>[RO]</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td><strong>Valid Error 0</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>[R/W]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Error: The error located in field 15:5 is valid.</td>
</tr>
</tbody>
</table>
### LBCFERRLOG43 - LBCF DPF Error log register 46

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:21</td>
<td><strong>Row Number for Error 1</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Row Number for Error1: The physical row Addr where the parity error has been detected in SLM Bank.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Slice 2 Bank 3 Error log register 00.</td>
</tr>
<tr>
<td></td>
<td>20:17</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>16</td>
<td>15:5</td>
<td><strong>Row Number for Error0</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Row Number for Error0: The physical row Addr where the parity error has been detected in SLM Bank.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Slice 2 Bank 3 Error log register 00.</td>
</tr>
<tr>
<td></td>
<td>4:1</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td><strong>Valid Error 0</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Error: The error located in field 15:5 is valid.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Slice 2 Bank 3 Error log register 00.</td>
</tr>
</tbody>
</table>
# LBCF DPF Error log register 47

## LBCFERRLOG44 - LBCF DPF Error log register 47

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
</table>
| 0     | 31:21 | **Row Number for Error 1**  
R/W | Row Number for Error1: The physical row Addr where the parity error has been detected in SLM Bank.  
Slice 2 Bank 3 Error log register 00. |
| 0     | 20:17 | **Reserved**  
RO |
| 16    | 15:5 | **Valid Error 0**  
R/W | Valid Error: The error located in field 31:21 is valid.  
Slice 2 Bank 3 Error log register 00. |
| 0     | 4:1  | **Reserved**  
RO |
| 0     | 0    | **Valid Error 0**  
R/W | Valid Error: The error located in field 15:5 is valid.  
Slice 2 Bank 3 Error log register 00. |
## LBS config bits

### LBSREG - LBS config bits

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x46000000 CHV, BSW</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0B124h</td>
</tr>
</tbody>
</table>

#### Config Bits for LBS unit

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:27</td>
<td>Retry timer for lookup into LSQC</td>
<td>01000b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Time between receiving Reject Response from LSQC and doing a snoop lookup request again onto LSQC.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>00000b: 0 clocks.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>00001b: 1 clocks.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>00010b: 2 clocks.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>01000b: 8 clocks (default value).</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>11111b: 32 clocks.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>lbcf_retry_timer[4:0].</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| 26    | | Recycle parent faster in R/W perf mode | 1b | R/W |
|-------|---------------------------------|-----------------|--------|
|       |Arc into recycle as soon as parent becomes eligible to be recycled. |
|       |0: Disabled (recycle possible only when parent is recycled). |
|       |1: Enabled (default). |
|       |lbcf_csr_lsqc_rwperf_quickrec. |

| 25    | | Perf mode for Writes to same address | 1b | |
|-------|---------------------------------|-----------------|--------|
|       |Performance improvement for writes to same address in L3: |
|       |0 - Performance mode is not enabled. |
|       |1 - Performance mode is enabled (default). |
|       |lbcf_csr_lsqc_eryrec. |

| 24:0  | Reserved | RO |

---

**Note:** The above table provides a detailed view of the LBS config bits, including their descriptions, default values, and access modes. This information is crucial for understanding the functionality and configuration of the LBS unit in the context of the projects CHV and BSW.
## LEAKAGECOUNTER

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>1300D4h</td>
</tr>
</tbody>
</table>

### Leakage counter readout

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Leakage Counter</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
</tbody>
</table>

This is a live read of the 32-bit leakage counter. Reading this register does not cause the counter to clear. Only the write-write protocol from Punit to Gunit will cause the leakage counter to clear.
## LEAKAGECOUNTERCTL

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>1300D0h</td>
</tr>
</tbody>
</table>

Leakage counter control.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31</td>
<td><strong>leakage_lock</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This is the lock bit for the leakage counter registers. 13_00D0-13_00DC.</td>
</tr>
<tr>
<td>30:1</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td><strong>leakage_count_en</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Enable the leakage counters. When zero, the counters will be held to zero. As this transitions to 0-1, counting is enabled. Like the EMON Energy Counters, the leakage counter will perform a &quot;sum of weights&quot; for the 16 events/subwells) described in 13_00D8 and 13_00DC.</td>
</tr>
</tbody>
</table>
**LEAKAGEWEIGHT1**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>1300D8h</td>
</tr>
</tbody>
</table>

Leakage weights for wells types 1 to 4.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value:</th>
<th>Access:</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>leakage_vdve</strong></td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Leakage weight for 'vdve' well type. One of these wells are expected on CHV, BSW.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23:16</td>
<td></td>
<td><strong>Reserved</strong></td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td>15:8</td>
<td></td>
<td><strong>Reserved</strong></td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td>7:0</td>
<td></td>
<td><strong>leakage_aon</strong></td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Leakage weight for 'aon' well type. One of these wells are expected on CHV, BSW.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# LEAKAGEWEIGHT2

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>leakage_eupair</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Leakage weight for 'eupair' well type. Eight of these wells are expected on CHV, BSW.</td>
</tr>
<tr>
<td>23:16</td>
<td><strong>leakage_ss</strong></td>
<td><strong>Default Value:</strong> 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Leakage weight for 'ss' well type. Two of these wells are expected on CHV, BSW.</td>
</tr>
<tr>
<td>15:8</td>
<td><strong>leakage_l3</strong></td>
<td><strong>Default Value:</strong> 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Leakage weight for 'l3' well type. One of these wells are expected on CHV, BSW.</td>
</tr>
<tr>
<td>7:0</td>
<td><strong>leakage_ffsc</strong></td>
<td><strong>Default Value:</strong> 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Leakage weight for 'ffsc' well type. One of these wells are expected on CHV, BSW.</td>
</tr>
</tbody>
</table>
## LNCF config save msg

<table>
<thead>
<tr>
<th><strong>Register Space</strong></th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Project</strong></td>
<td>CHV, BSW</td>
</tr>
<tr>
<td><strong>Source</strong></td>
<td>PRM</td>
</tr>
<tr>
<td><strong>Default Value</strong></td>
<td>0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits)</strong></td>
<td>32</td>
</tr>
<tr>
<td><strong>Address</strong></td>
<td>0B0FCh</td>
</tr>
</tbody>
</table>

This register is not context saved and is written by pm unit.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:10</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td>9:0</td>
<td></td>
<td><strong>Context save bit</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W Hardware Clear</td>
</tr>
</tbody>
</table>

- **Bit[9].Power Context Save Request**
  - 0: Power context save is not being requested (default).
  - 1: Power context save is being requested.
  - Unit needs to self-clear this bit upon sampling.
- **Bits[8:0].QWord Credits for Power Context Save Request**
  - Minimum Credits = 1: Unit may send 1 QWord pair (enough for first LRI at least).
  - Maximum Credits = 511: Unit may send 511 QWord pairs.
  - A QWord pair is defined as a 32-bit register address and the corresponding 32-bits of register data. Note that the LRI header and END commands are 64-bits each (32-bit command followed by 32-bit NOOP) and consume one QWord credit.
  - Only valid with PWRCTX_SAVE_REQ (Bit9).
Load Indirect Base Vertex

<table>
<thead>
<tr>
<th>3DPRIM_BASE_VERTEX - Load Indirect Base Vertex</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: RenderCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Access: R/W</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 02440h-02443h</td>
</tr>
<tr>
<td>Valid Projects:</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Base Vertex</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: S31</td>
</tr>
</tbody>
</table>
|       |     | This register is used to store the Base Vertex of the 3D_PRIMITIVE command when Load Indirect Enable is set.
Load Indirect Instance Count

<table>
<thead>
<tr>
<th>3DPRIM_INSTANCE_COUNT - Load Indirect Instance Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: RenderCS</td>
</tr>
<tr>
<td>Default Value: 0x000000000</td>
</tr>
<tr>
<td>Access: R/W</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 02438h-0243Bh</td>
</tr>
<tr>
<td>Valid Projects:</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Instance Count</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>This register is used to store the Instance Count of the 3D_PRIMITIVE command when Load Indirect Enable is set.</td>
</tr>
</tbody>
</table>
## Load Indirect Start Instance

### 3DPRIM_START_INSTANCE - Load Indirect Start Instance

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>RenderCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x000000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0243Ch-0243Fh</td>
</tr>
<tr>
<td>Valid Projects:</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Start Vertex</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: U32</td>
</tr>
</tbody>
</table>

This register is used to store the Start Instance of the 3D_PRIMITIVE command when Load Indirect Enable is set.
# Load Indirect Start Vertex

<table>
<thead>
<tr>
<th><strong>3DPRIM_START_VERTEX - Load Indirect Start Vertex</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: RenderCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Access: R/W</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 02430h-02433h</td>
</tr>
<tr>
<td>Valid Projects:</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>DWord</strong></th>
<th><strong>Bit</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Start Vertex</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: U32</td>
</tr>
</tbody>
</table>

This register is used to store the Start Vertex of the 3D_PRIMITIVE command when Load Indirect Enable is set.
**Load Indirect Vertex Count**

<table>
<thead>
<tr>
<th>3DPRIM_VERTEX_COUNT - Load Indirect Vertex Count</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> RenderCS</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 02434h-02437h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Valid Projects:</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>DWord</td>
<td>Bit</td>
</tr>
<tr>
<td>--------</td>
<td>-----</td>
</tr>
<tr>
<td>0</td>
<td>31:0</td>
</tr>
</tbody>
</table>

Format: U32

This register is used to store the Vertex Count of the 3D_PRIMITIVE command when Load Indirect Enable is set.
# LOW 2X FREQUENCY THRESHOLD

<table>
<thead>
<tr>
<th>LOW2XFREQTHRESH - LOW 2X FREQUENCY THRESHOLD</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x000000A0</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 101128h</td>
</tr>
<tr>
<td><strong>New for CHV, BSW.</strong></td>
</tr>
<tr>
<td>BIOS/Driver programs the Low2x frequency threshold.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td><strong>RESERVED</strong></td>
<td>0000h</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15:0</td>
<td>L3FREQREG</td>
<td><strong>L3FREQREG</strong></td>
<td>00A0h</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access:</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

This is control signal needed from clock unit that can be set at 1 when 2X clock frequency is less than or equal to 1GHz. It needs to be at 0 when 2X clock frequency is > 1GHz.

Value when in HPLL mode: low2xthresh=0x00B4. This sets low2xfreq high whenever cu2x frequency is less than or equal to 800, 800, 1000, 800, and 800 for CZ400, CZ320, CZ333, CZ266, and CZ200 respectively.

Value when in GPLL mode: low2xthresh=0x00A0. This sets low2xfreq high whenever cu2x frequency is less than or equal to 1000, 800, 834, 889, and 800 for CZ400, CZ320, CZ333, CZ266, and CZ200 respectively.
# LPFC control register

<table>
<thead>
<tr>
<th><strong>Register</strong></th>
<th><strong>Value</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register</strong></td>
<td><strong>Value</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td><strong>Register</strong></td>
<td><strong>Value</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td><strong>Register</strong></td>
<td><strong>Value</strong></td>
<td><strong>Description</strong></td>
</tr>
</tbody>
</table>

## LPFC control register

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31</td>
<td>LPFC enable signal</td>
</tr>
<tr>
<td></td>
<td>30:0</td>
<td>LPPC event collection enable signal. Incf_lpfc_cnt_en.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td>Reserved.</td>
<td></td>
</tr>
</tbody>
</table>
### LTCD Error Injection Register

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:25</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td>31:25</td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td>31:25</td>
<td>Access: RO</td>
</tr>
<tr>
<td>24</td>
<td></td>
<td><strong>LTISEQSL parity error interrupt</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Parity error interrupt to LTISEQ Slice.</td>
</tr>
<tr>
<td>23</td>
<td></td>
<td><strong>Bank hang on parity disable</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td>22</td>
<td></td>
<td><strong>Parity Error Injection Enable</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0: Disable parity error injection.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1: Enable parity error injection.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>lbcf_parity_err_inject_en.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Do not Enable this when ECC Error injection is enabled.</td>
</tr>
<tr>
<td>21</td>
<td></td>
<td><strong>Double Bit Error injection</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0: Default No error injected.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1: Double bit error is injected.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>lbcf_ecc_2bit_err_inject.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Single bit Error Injection and double bit error injection are mutually exclusive.</td>
</tr>
<tr>
<td>Register Name</td>
<td>Description</td>
<td>Project</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
<td>---------</td>
</tr>
<tr>
<td>LBCFERR - LTCD Error Injection Register</td>
<td>Do not Enable this when Parity Error injection is enabled.</td>
<td></td>
</tr>
</tbody>
</table>
| 20 Single Bit Error injection | | CHV, BSW | R/W | 0: Default No error injected. 1: Single bit error is injected.  
| | | | | lbcf_ecc_1bit_err_inject.  
| | | | | Single bit Error Injection and double bit error injection are mutually exclusive.  
| | | | | Do not Enable this when Parity Error injection is enabled. |
| 19 ECC Error Injection Enable | | CHV, BSW | R/W | 0: Disable ECC error injection. 1: Enable ECC error injection.  
| | | | | lbcf_ecc_err_inject_en.  
| | | | | Do not Enable this when Parity Error injection is enabled. |
| 18:4 Row address for error injection | | CHV, BSW | R/W | Row address for which error is injected. For ECC error injection, the same row address is overloaded to inject ECC error for that particular row address.  
| | | | | SLM configuration.  
| | | | | For Address: Rowaddress [16:7] is applicable for SLM banks and Parity injection is enabled and Rowaddress[18:7] is applicable for ECC (for both NonSLM/SLM banks).  
| | | | | Sub bank is applicable only for Parity injection. Only Subbank 0, 1, 2, 3 are supported and are only related to SLM Banks (4KB).  
| | | | | Total 12 bits address.  
| | | | | Bits 18:7: Row address.  
| | | | | Bits 6:4: Sub-Bank number.  
| | | | | lbcf_parity_err_inject_rowaddr[9:0],lbcf_parity_err_inject_subank[2:0],16:7 - For SLM parity only.  
| | | | | lbcf_parity_err_inject_rowaddr[9:0],lbcf_ecc_err_inject_rowaddr[11:0].For ECC error injection, the same row address is overloaded to inject ECC error for that particular row address. |
| 3:2 Bank ID for error injection | | CHV, BSW | R/W | | 00b: Inject in Bank 0. 01b: Inject in Bank 1. 10b: Inject in Bank 2. 11b: Inject in Bank 3 (error injection not supported for SLM bank3). |
### LBCFERR - LTCD Error Injection Register

<table>
<thead>
<tr>
<th>1:0</th>
<th>Slice ID for Error Injection</th>
</tr>
</thead>
<tbody>
<tr>
<td>lbcf_err_inject_bankid[1:0].</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Project: CHV, BSW</th>
<th>Access: R/W</th>
</tr>
</thead>
<tbody>
<tr>
<td>00b:</td>
<td>Inject error in Slice 0.</td>
<td></td>
</tr>
<tr>
<td>01b:</td>
<td>Inject error in Slice 1.</td>
<td></td>
</tr>
<tr>
<td>10b:</td>
<td>Inject error in Slice 2.</td>
<td></td>
</tr>
<tr>
<td>lbcf_err_inject_sliceid.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## MA - MA

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>PCI: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>00094h</td>
</tr>
</tbody>
</table>

### Message Address

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:2</td>
<td>ADDRESS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MA: Lower 32-bits of the system specified message address, always DW aligned. When GVD issues an MSI interrupt as a MEMWR on the SCL, the memory address corresponds to the value of this field.</td>
</tr>
<tr>
<td>1:0</td>
<td>RESERVED</td>
<td>Default Value:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
</tbody>
</table>
Main Graphic Arbiter Error Report

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>040A0h</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31</td>
<td>Reserved Error Bits 31</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td>30</td>
<td>30</td>
<td>Reserved Error Bits 30</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td>29</td>
<td>29</td>
<td>Reserved Error Bits 29</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td>28</td>
<td>28</td>
<td>Reserved Error Bits 28</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td>27</td>
<td>27</td>
<td>Reserved Error Bits 27</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td>26</td>
<td>26</td>
<td>Reserved Error Bits 26</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>
## ERROR - Main Graphic Arbiter Error Report

<table>
<thead>
<tr>
<th></th>
<th>Future Use.</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td><strong>Reserved Error Bits 25</strong></td>
</tr>
<tr>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td>Future Use.</td>
</tr>
<tr>
<td>24</td>
<td><strong>Reserved Error Bits 24</strong></td>
</tr>
<tr>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td>Future Use.</td>
</tr>
<tr>
<td>23</td>
<td><strong>Reserved Error Bits 23</strong></td>
</tr>
<tr>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td>Future Use.</td>
</tr>
<tr>
<td>22</td>
<td><strong>Reserved Error Bits 22</strong></td>
</tr>
<tr>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td>Future Use.</td>
</tr>
<tr>
<td>21</td>
<td><strong>Reserved Error Bits 21</strong></td>
</tr>
<tr>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td>Future Use.</td>
</tr>
<tr>
<td>20</td>
<td><strong>Reserved Error Bits 20</strong></td>
</tr>
<tr>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td>Future Use.</td>
</tr>
<tr>
<td>19</td>
<td><strong>Reserved Error Bits 19</strong></td>
</tr>
<tr>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td>Future Use.</td>
</tr>
<tr>
<td>18</td>
<td><strong>Reserved Error Bits 18</strong></td>
</tr>
<tr>
<td></td>
<td>Default Value: 0b</td>
</tr>
</tbody>
</table>
## ERROR - Main Graphic Arbiter Error Report

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Access:</strong></td>
<td><strong>R/W</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Future Use.</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Reserved Error Bits 17

<table>
<thead>
<tr>
<th>Default Value</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>0b</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Access:**

<table>
<thead>
<tr>
<th><strong>R/W</strong></th>
<th></th>
</tr>
</thead>
</table>

### Reserved Error Bits 16

<table>
<thead>
<tr>
<th>Default Value</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>0b</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Access:**

<table>
<thead>
<tr>
<th><strong>R/W</strong></th>
<th></th>
</tr>
</thead>
</table>

### Reserved Error Bits 15

<table>
<thead>
<tr>
<th>Default Value</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>0b</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Context:**

| **ctx_fault_ctxt_not_prsmt_err** | The Present (P) field in the context-entry used to process the DMA request is Clear. |

**Access:**

<table>
<thead>
<tr>
<th><strong>R/W</strong></th>
<th></th>
</tr>
</thead>
</table>

### Reserved Error Bits 14

<table>
<thead>
<tr>
<th>Default Value</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>0b</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Context:**

| **ctx_fault_root_not_prsmt_err** | The present (UP/LP) field in the root-entry used to process the untranslated request with PASID is 0. |

**Access:**

<table>
<thead>
<tr>
<th><strong>R/W</strong></th>
<th></th>
</tr>
</thead>
</table>

### Reserved Error Bits 13

<table>
<thead>
<tr>
<th>Default Value</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>0b</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Context:**

| **ctx_fault_pasid_not_prsnt_err** | PASID Table entry to be used does not have the PRESENT flag set. This means the PASID entry is not valid. |

**Access:**

<table>
<thead>
<tr>
<th><strong>R/W</strong></th>
<th></th>
</tr>
</thead>
</table>

### Reserved Error Bits 12

<table>
<thead>
<tr>
<th>Default Value</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>0b</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Context:**

| **ctx_fault_pasid_ovflw_err** | PASID Table size in extended context entry defines the number of PASIDs that will be supported. If hardware receives a PASID number outside the supported boundary, report as an error. |

**Access:**

<table>
<thead>
<tr>
<th><strong>R/W</strong></th>
<th></th>
</tr>
</thead>
</table>

### Reserved Error Bits 11

<table>
<thead>
<tr>
<th>Default Value</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>0b</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Context:**

| **ctx_fault_root_not_prsmt_err** | The present (UP/LP) field in the root-entry used to process the untranslated request with PASID is 0. |

**Access:**

<table>
<thead>
<tr>
<th><strong>R/W</strong></th>
<th></th>
</tr>
</thead>
</table>

**Context:**

| **ctx_fault_ctxt_not_prsmt_err** | The Present (P) field in the context-entry used to process the DMA request is Clear. |

**Access:**

<table>
<thead>
<tr>
<th><strong>R/W</strong></th>
<th></th>
</tr>
</thead>
</table>
## ERROR - Main Graphic Arbiter Error Report

<table>
<thead>
<tr>
<th>Bit Number</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>Reserved Error Bits 10</td>
<td>0b</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Reserved</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Unloaded PD Error</td>
<td>0b</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Reserved Error Bits 7</td>
<td>0b</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Reserved</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Reserved</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Reserved</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Reserved</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Invalid Page Directory Entry Error</td>
<td>0b</td>
<td>R/W</td>
<td>PD entry's valid bit is 0.</td>
</tr>
<tr>
<td>1</td>
<td>Reserved</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>TLB Page Fault Error</td>
<td>0b</td>
<td>R/W</td>
<td>A TLB Page's GTT translation generated a page fault (GTT entry not valid).</td>
</tr>
</tbody>
</table>

- **ctx_fault_pasid_dis_err** - Submission of advanced context where the PASID field is not enabled in the extended context entry.
- **rstrm_fault_nowb_atomic_err** - All page table accesses in advanced context with A/D bits are considered as atomic operations in WB space. However if the memory type for the page table accesses come out as anything but WB, that is an error.
## Main Graphic Arbiter Error Report 2

<table>
<thead>
<tr>
<th><strong>ERROR_2 - Main Graphic Arbiter Error Report 2</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 040A4h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>DWord</strong></th>
<th><strong>Bit</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Main Graphic Arbiter Error Report 2</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit [5:0] - tlbpend_reg_faultcnt[5:0].</td>
</tr>
</tbody>
</table>
Main Graphic Arbiter Error Report 3

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Project:</th>
<th>Source:</th>
<th>Default Value:</th>
<th>Access:</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td>Reserved</td>
<td>CHV, BSW</td>
<td>PRM</td>
<td>0000h</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>Error3 Error Bits 15</td>
<td></td>
<td></td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td>Error3 Error Bits 14</td>
<td></td>
<td></td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
<td>Error3 Error Bits 13</td>
<td></td>
<td></td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>Error3 Error Bits 12</td>
<td></td>
<td></td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>Error3 Error Bits 11</td>
<td></td>
<td></td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

This register is used to report different error conditions. Error bits are writable.
### ERROR_3 - Main Graphic Arbiter Error Report 3

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Project</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>Error3 Error Bits 10</td>
<td>0b</td>
<td>CHV, BSW</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>Project: invalid_varmtrr_overlap_memtype_wr.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Error3 Error Bits 9</td>
<td>0b</td>
<td>CHV, BSW</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>Project: invalid_default_memtype_value_rd.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Error3 Error Bits 8</td>
<td>0b</td>
<td>CHV, BSW</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>Project: invalid_default_memtype_value_wr.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Error3 Error Bits 7</td>
<td>0b</td>
<td>CHV, BSW</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>Project: invalid_varmtrr_memtype_value_rd.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Error3 Error Bits 6</td>
<td>0b</td>
<td>CHV, BSW</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>Project: invalid_varmtrr_memtype_value_wr.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Error3 Error Bits 5</td>
<td>0b</td>
<td>CHV, BSW</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>Project: invalid_fixedmtrr_memtype_value_rd.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# ERROR_3 - Main Graphic Arbiter Error Report 3

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Project</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Error3 Error Bits 3</td>
<td>0b</td>
<td>CHV, BSW</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>invalid_fixedmtrr_memtype_value_wr.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Reserved</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Error3 Error Bits 1</td>
<td>0b</td>
<td>CHV, BSW</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>gttc_internal_error.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>Error3 Error Bits 0</td>
<td>0b</td>
<td>CHV, BSW</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>reg_wrid_internal_error.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# Main Graphic Arbiter Error Report Register

**GFX_ARB_ERROR_RPT - Main Graphic Arbiter Error Report Register**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>RenderCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Trusted Type:</td>
<td>1</td>
</tr>
<tr>
<td>Address:</td>
<td>040A0h</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format:</strong> MBZ</td>
</tr>
<tr>
<td>15:9</td>
<td><strong>Reserved</strong></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td><strong>Reserved</strong></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td><strong>Reserved</strong></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td><strong>Reserved</strong></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td><strong>Reserved</strong></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td><strong>Reserved</strong></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td><strong>Hardware Status Page Fault Error</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>HWSP's GTT translation generated a page fault (GTT entry not valid).</strong></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td><strong>Invalid Page Directory entry error</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>PD entry's valid bit is 0.</strong></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td><strong>Context Page Fault Error</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>A Context Page's GTT translation generated a page fault (GTT entry not valid).</strong></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td><strong>TLB Page Fault Error</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>A TLB Page's GTT translation generated a page fault (GTT entry not valid).</strong></td>
<td></td>
</tr>
</tbody>
</table>
**MASTER_INT_CTL**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>44200h</td>
</tr>
</tbody>
</table>

This register has the master enable for graphics interrupts and gives an overview of what interrupts are pending. An interrupt pending bit will read 1b while one or more interrupts of that category are set (IIR) and enabled (IER). All Pending Interrupts are ORed together to generate the combined interrupt. The combined interrupt is ANDed with the Master Interrupt enable to create the master enabled interrupt. The master enabled interrupt goes to PCI device 2 interrupt processing. The master interrupt enable must be set before any of these interrupts will propagate to PCI device 2 interrupt processing.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31</td>
<td><strong>MASTER_INTERRUPT_ENABLE</strong></td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>THIS IS THE MASTER CONTROL FOR GRAPHICS INTERRUPT. This must be enabled for any of these interrupts to propagate to PCI dev2 interrupt processing. 0b - master interrupt disable, 1b - master interrupt enable</td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td></td>
<td><strong>PCU_INTERRUPT_PENDING</strong></td>
<td>0b</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This field indicates if interrupts of this category is pending.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>29:8</td>
<td></td>
<td><strong>Reserved</strong></td>
<td>00000b</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td><strong>WDBOX_OACS_INTERRUPT_PENDING</strong></td>
<td>0b</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This field indicates if interrupts of this category is pending.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td><strong>VEBOX_INTERRUPT_PENDING</strong></td>
<td>0b</td>
<td>RO</td>
</tr>
</tbody>
</table>

---

**Command Reference: Registers**

**Doc Ref # IHD-OS-CHV-BSW-Vol 2c-10.15**
# MASTER_INT_CTL - MASTER_INT_CTL

<table>
<thead>
<tr>
<th>Access</th>
<th>RO</th>
</tr>
</thead>
<tbody>
<tr>
<td>This field indicates if interrupts of this category is pending.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Access</th>
<th>RO</th>
</tr>
</thead>
<tbody>
<tr>
<td>This field indicates if interrupts of this category is pending.</td>
<td></td>
</tr>
</tbody>
</table>

| Reserved     |    |

| Default Value | 0b |
| Access        | RO |
| This field indicates if interrupts of this category is pending. |    |

| Default Value | 0b |
| Access        | RO |
| This field indicates if interrupts of this category is pending. |    |

| Default Value | 0b |
| Access        | RO |
| This field indicates if interrupts of this category is pending. |    |

| Default Value | 0b |
| Access        | RO |
| This field indicates if interrupts of this category is pending. |    |

| Default Value | 0b |
| Access        | RO |
| This field indicates if interrupts of this category is pending. |    |
## Master start timer

### MASTIMER - Master start timer

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x000000001</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0B438h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Master start timer value</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000000000000000000000000001b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

Master Start Timer (MSTSTTMR).

`lpconf lpfc_master_start_timer [31:0]`. So many clocks are expired before starting the rest of the counters. Time to wait is 256 * value clocks. Value for this register cannot be 0.
# GFX_PEND_TLB_0 - Max Outstanding Pending TLB Requests 0

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>04034h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31</td>
<td><strong>TEX Limit Enable Bit</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This bit is used to enable the pending TLB requests limitation function for the Texture Cache. When set, the number of internal pending read requests which require a TLB read does not exceed the programmed counter value.</td>
</tr>
<tr>
<td>30</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>29:24</td>
<td></td>
<td><strong>TEX TLB Limit Count</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This is the MAX number of Allowed internal pending read requests which require a TLB read.</td>
</tr>
<tr>
<td>23</td>
<td></td>
<td><strong>DC Limit Enable Bit</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This bit is used to enable the pending TLB requests limitation function for the Instruction Cache. When set, the number of internal pending read requests which require a TLB read does not exceed the programmed counter value.</td>
</tr>
<tr>
<td>22</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>21:16</td>
<td></td>
<td><strong>DC TLB Limit Count</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This is the MAX number of Allowed internal pending read requests which require a TLB read.</td>
</tr>
</tbody>
</table>
### GFX_PEND_TLB_0 - Max Outstanding Pending TLB Requests

<table>
<thead>
<tr>
<th>Offset</th>
<th>Field</th>
<th>Default Value</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td><strong>VF Limit Enable Bit</strong></td>
<td>0b</td>
<td>R/W</td>
<td>This bit is used to enable the pending TLB requests limitation function for the Vertex Fetch. When set, the number of internal pending read requests which require a TLB read does not exceed the programmed counter value.</td>
</tr>
<tr>
<td>4</td>
<td><strong>Reserved</strong></td>
<td>0b</td>
<td>RO</td>
<td></td>
</tr>
<tr>
<td>6:0</td>
<td><strong>VF TLB Limit Count</strong></td>
<td>000000b</td>
<td>R/W</td>
<td>This is the MAX number of Allowed internal pending read requests which require a TLB read.</td>
</tr>
<tr>
<td>10</td>
<td><strong>VMC Limit Enable bit</strong></td>
<td>0b</td>
<td>R/W</td>
<td>This bit is used to enable the pending TLB requests limitation function for the Video Motion Compensation. When set, the number of internal pending read requests which require a TLB read does not exceed the programmed counter value.</td>
</tr>
<tr>
<td>14</td>
<td><strong>Reserved</strong></td>
<td>0b</td>
<td>RO</td>
<td></td>
</tr>
<tr>
<td>12:8</td>
<td><strong>VMC TLB Limit Count</strong></td>
<td>000000b</td>
<td>R/W</td>
<td>This is the MAX number of Allowed internal pending read requests which require a TLB read.</td>
</tr>
</tbody>
</table>
Max Outstanding Pending TLB Requests 1

<table>
<thead>
<tr>
<th>GFX_PEND_TLB_1 - Max Outstanding Pending TLB Requests 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 04038h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31</td>
<td>SOL Limit Enable Bit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This bit is used to enable the pending TLB requests limitation function for the SOL. When set, the number of internal pending read requests which require a TLB read does not exceed the programmed counter value.</td>
</tr>
<tr>
<td>30</td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>29:24</td>
<td></td>
<td>SOL TLB Limit Count</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This is the MAX number of Allowed internal pending read requests which require a TLB read.</td>
</tr>
<tr>
<td>23</td>
<td></td>
<td>L3 Limit Enable Bit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This bit is used to enable the pending TLB requests limitation function for the L3. When set, the number of internal pending read requests which require a TLB read does not exceed the programmed counter value.</td>
</tr>
<tr>
<td>22</td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>21:16</td>
<td></td>
<td>L3 TLB Limit Count</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This is the MAX number of Allowed internal pending read requests which require a TLB read.</td>
</tr>
<tr>
<td></td>
<td>Description</td>
<td>Default Value</td>
</tr>
<tr>
<td>---</td>
<td>--------------------------------------------------------------------------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>15</td>
<td><strong>RCZ Limit Enable Bit</strong></td>
<td>0b</td>
</tr>
<tr>
<td>14</td>
<td><strong>Reserved</strong></td>
<td>0b</td>
</tr>
<tr>
<td>13:8</td>
<td><strong>RCZ TLB Limit Count</strong></td>
<td>000000b</td>
</tr>
<tr>
<td>7</td>
<td><strong>RCC Limit Enable bit</strong></td>
<td>0b</td>
</tr>
<tr>
<td>6</td>
<td><strong>Reserved</strong></td>
<td>0b</td>
</tr>
<tr>
<td>5:0</td>
<td><strong>RCC TLB Limit Count</strong></td>
<td>000000b</td>
</tr>
</tbody>
</table>
MAX Requests Allowed - GAM

<table>
<thead>
<tr>
<th>GFX_MAX_REQ_COUNT - MAX Requests Allowed - GAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x43F20101</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 04AA4h</td>
</tr>
</tbody>
</table>

Programmable Request Count - GAM

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:26</td>
<td>GAP Writes Max Request Limit Count</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 010000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This is the MAX number of Allowed Write Requests Count - These counters keep track of the accepted write requests from all GAP clients (RCZ, HiZ, Stc, RCC, L3). Minimum count value must be = 1.</td>
</tr>
<tr>
<td>25:20</td>
<td></td>
<td>CVS Max Request Limit Count</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 111111b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This is the MAX number of Allowed Requests Count - These counters keep track of the accepted requests from each client. Requests are counted, regardless of kind of cycle (Miss/Hit/Present). Minimum count value must be = 1.</td>
</tr>
<tr>
<td>19</td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>18:13</td>
<td></td>
<td>L3 Max Request Limit Count</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 010000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This is the MAX number of Allowed Requests Count - These counters keep track of the accepted requests from each client. Requests are counted, regardless of kind of cycle (Miss/Hit/Present). Minimum count value must be = 1.</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
<td>Default Value</td>
</tr>
<tr>
<td>---------</td>
<td>--------------------------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>11:6</td>
<td>Z Request Limit Count</td>
<td>000100b</td>
</tr>
<tr>
<td></td>
<td>This is the MAX number of Allowed Requests Count - These counters keep track of the accepted requests from each client. Requests are counted, regardless of kind of cycle (Miss/Hit/Present). Minimum count value must be = 1.</td>
<td></td>
</tr>
<tr>
<td>5:0</td>
<td>RCC Request Limit Count</td>
<td>000001b</td>
</tr>
<tr>
<td></td>
<td>This is the MAX number of Allowed Requests Count - These counters keep track of the accepted requests from each client. Requests are counted, regardless of kind of cycle (Miss/Hit/Present). Minimum count value must be = 1.</td>
<td></td>
</tr>
</tbody>
</table>
# MAX Requests Allowed - MFX

## MEDIA_MAX_REQ_COUNT - MAX Requests Allowed - MFX

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x10201020</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>04AA0h</td>
</tr>
</tbody>
</table>

### Programmable Request Count - MFX

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>GFX Max Request Limit Count</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00010000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This is the MAX number of Allowed Requests Count - These counters keep track of the accepted requests from each engine. Requests are counted, regardless of kind of cycle (Miss/Hit/Present). Minimum count value must be = 1.</td>
</tr>
<tr>
<td>0</td>
<td>23:16</td>
<td><strong>MFX Max Request Limit Count</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00100000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This is the MAX number of Allowed Requests Count - These counters keep track of the accepted requests from each engine. Requests are counted, regardless of kind of cycle (Miss/Hit/Present). Minimum count value must be = 1.</td>
</tr>
<tr>
<td>0</td>
<td>15:14</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>0</td>
<td>13:8</td>
<td><strong>VLF Max Request Limit Count</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 010000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This is the MAX number of Allowed Requests Count - These counters keep track of the accepted requests from each client. Requests are counted, regardless of kind of cycle (Miss/Hit/Present). Minimum count value must be = 1.</td>
</tr>
<tr>
<td>0</td>
<td>7:6</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
</tbody>
</table>
### MEDIA_MAX_REQ_COUNT - MAX Requests Allowed - MFX

<table>
<thead>
<tr>
<th>5:0</th>
<th><strong>MFX Max Request Limit Count</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Default Value:</strong> 100000b</td>
</tr>
<tr>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
</tbody>
</table>

This is the MAX number of Allowed Requests Count - These counters keep track of the accepted requests from each client. Requests are counted, regardless of kind of cycle (Miss/Hit/Present). Minimum count value must be = 1.
MAX Requests Allowed - VEBX and BLT

**VEBX_BLIT_MAX_REQ_COUNT - MAX Requests Allowed - VEBX and BLT**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x08081020</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>04AA8h</td>
</tr>
</tbody>
</table>

Programmable Request Count - VEBX and BLT

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>BLT Max Request Limit Count</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00001000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This is the MAX number of Allowed Requests Count - These counters keep track of the accepted requests from each engine. Requests are counted, regardless of kind of cycle (Miss/Hit/Present).</td>
</tr>
<tr>
<td>23:16</td>
<td>VEBX Max Request Limit Count</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default Value: 00001000b</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td>This is the MAX number of Allowed Requests Count - These counters keep track of the accepted requests from each engine. Requests are counted, regardless of kind of cycle (Miss/Hit/Present).</td>
<td></td>
</tr>
<tr>
<td>15:8</td>
<td>VLF1 Max Request Limit Count</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default Value: 00010000b</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td>This is the MAX number of Allowed Requests Count - These counters keep track of the accepted requests from each engine. Requests are counted, regardless of kind of cycle (Miss/Hit/Present).</td>
<td></td>
</tr>
<tr>
<td>7:0</td>
<td>MFX1 Max Request Limit Count</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default Value: 00100000b</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td>This is the MAX number of Allowed Requests Count - These counters keep track of the accepted requests from each engine. Requests are counted, regardless of kind of cycle (Miss/Hit/Present).</td>
<td></td>
</tr>
</tbody>
</table>
## MBC Control Register

### MBCTL - MBC Control Register

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:17</td>
<td><strong>ECORSVD</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ECO purposes Reserved</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td><strong>Fuse Write to VCR as Nonposted</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 - Default - MBCunit will send the VCR (Fuse) writes as Posted Write cycles</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 - MBCunit will send the VCR (Fuse) writes as Non Posted Write cycles</td>
</tr>
<tr>
<td>15:8</td>
<td></td>
<td><strong>RSVD</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RSVD</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td><strong>Disable Wait for G3d Outbound empty in MAE</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 - Wait for SQempty for MAE update Flow</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 - MBC MAE update FSM will not wait for the SQempty to complete the FSM</td>
</tr>
<tr>
<td>6:5</td>
<td></td>
<td><strong>RSVD</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RSVD</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td><strong>RSVD</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RSVD</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td><strong>RSVD</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RSVD</td>
</tr>
<tr>
<td></td>
<td>MBCTL - MBC Control Register</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>-----------------------------</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td><strong>BME Update Enable</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td>BME update Enable -</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 - Default Mae Update is not Enable. MBC will ignore all the BME updates from SA.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 - BME update is Enabled.</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td><strong>MAE Update Enable</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MAE update Enable -</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 - Default Mae Update is not Enable. MBC will ignore all the MAE updates from SA.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 - MAE update is Enabled. MBC will respond to the MAE updates.</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td><strong>RSVD</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access: RO</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 = Use MBC to GAM path with 8B reads (default)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 = Use MBC to Config Agent register interface with 4B reads</td>
<td></td>
</tr>
</tbody>
</table>
**MBDSM**

### MBDSM - MBDSM

<table>
<thead>
<tr>
<th>Register Space</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits)</td>
<td>32</td>
</tr>
<tr>
<td>Address</td>
<td>1080C0h</td>
</tr>
</tbody>
</table>

Mirror of base of data stolen memory. Simply a read-only alias to the configuration registers. Used by the MBC for boot context fetches.

This register contains the base address of Graphics Data Stolen DRAM memory. Note: This register is in device 0, 0xB0. Mirrored into device2, 0x5C. Graphics Stolen Memory is within DRAM space. The base of stolen memory will always be below 4G.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:20</td>
<td><strong>BDSM</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
</tbody>
</table>

**BDSM: BASE_OF_Data_STOLEN_MEMORY.**

This register contains bits 31 to 20 of the base address of Data stolen DRAM memory. For certain GTLC generated accesses, this base register will be added to GTLC provided offset address, forming the full physical address for the PFI fabric. This is also used as a base for VGA paged accesses.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>19:1</td>
<td></td>
<td><strong>RESERVED</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
</tbody>
</table>

Reserved

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td><strong>BDSM_LOCK</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
</tbody>
</table>

When set to 1b, this bit will lock all the bits in this register, including itself.
Command Reference: Registers

## MBGSM

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>108100h</td>
</tr>
</tbody>
</table>

Mirror of base of graphics stolen memory. Simply a read-only alias to the configuration registers. Used by the MBC for boot context fetches.

Base of GTT table in Gfx Stolen Memory

Note: This register is located in device 0, 0xB4. Mirrored into Device 2.

The GTT table is located within Graphics Stolen Memory in DRAM space.

The base of stolen memory will always be below 4G.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:20</td>
<td><strong>BGSM</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
</tbody>
</table>

BGSM: Gfx Base of GTT Stolen Memory. This register contains bits 31 to 20 of the base address of GTT Table in stolen DRAM memory.

BIOS determines base of GTT stolen memory by subtracting the GTT graphics stolen memory size (PCI offset 50 bits 9:8) from the Graphics Base of Data stolen (PCI offset 5C bits 31:20).

<table>
<thead>
<tr>
<th>19:1</th>
<th>RESERVED</th>
<th>Default Value: 00000h</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Access:</td>
<td>RO</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>0</th>
<th><strong>BGSM_LOCK</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Value:</td>
<td>0b</td>
</tr>
<tr>
<td>Access:</td>
<td>RO</td>
</tr>
</tbody>
</table>

This bit will lock all writeable settings in this register including itself.
### MD - MD

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>PCI: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>00098h</td>
</tr>
</tbody>
</table>

#### Message Data

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td>RESERVED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td>15:0</td>
<td>DATA</td>
<td>Default Value: 0000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

MD: This 16-bit field is programmed by system software. This forms the lower word of data for the MSI write transaction.
MEDFW_ACK

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td>RESERVED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved.</td>
</tr>
<tr>
<td>15:0</td>
<td>FWAKEMEDIAACK</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default Value: 0000h</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access: RO</td>
<td></td>
</tr>
</tbody>
</table>

This register contains the per thread force wake acknowledge bits for the Media power well.

For example, if 13_00B8[0] is written to a 1 (along with 13_00B8[16]=’1’), then bit0 of this register indicates when the force wake request has been completed.
## Media 1 TLB Control Register

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:1</td>
<td><strong>Reserved</strong></td>
<td>00000000000000000000000000000000b</td>
<td>RO</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td><strong>Invalidate TLBs on the corresponding Engine</strong></td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

SW writes 1 to invalidate the TLBs for the associated engine and HW clears the bit when invalidation is complete. To ensure proper invalidation of the TLBs, SW has to ensure the corresponding engine's HW pipeline is flushed and cleared from all its memory accesses. Otherwise HW cannot guarantee the proper invalidation for TLBs.

This bit is self clear.
# Media 2 TLB Control Register

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Reserved</strong></td>
<td>00000000000000000000000000000000b</td>
<td>RO</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td><strong>Invalidate TLBs on the corresponding Engine</strong></td>
<td>0b</td>
<td>R/W</td>
</tr>
</tbody>
</table>

- **Register Space:** MMIO: 0/2/0
- **Project:** CHV, BSW
- **Source:** PRM
- **Default Value:** 0x00000000
- **Size (in bits):** 32

**Address:** 04268h

- SW writes 1 to invalidate the TLBs for the associated engine and HW clears the bit when invalidation is complete. To ensure proper invalidation of the TLBs, SW has to ensure the corresponding engine's HW pipeline is flushed and cleared from all its memory accesses. Otherwise HW cannot guarantee the proper invalidation for TLBs.

This bit is self clear.
Media forcewake request

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td>FWAKEMEDIAREQMSK</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Mask bits for lower 16 bits to avoid a read modify/write. If '0', the corresponding bit in [15:0] is not changed. If '1', the corresponding bit in [15:0] is changed to the value in [15:0]</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>FWAKEMEDIAREQ15</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Force Wake Media request 15.</td>
</tr>
<tr>
<td>14</td>
<td></td>
<td>FWAKEMEDIAREQ14</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Force Wake Render request 14.</td>
</tr>
<tr>
<td>13</td>
<td></td>
<td>FWAKEMEDIAREQ13</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

This register contains per thread force wake request bits for the Render power Well. The upper 16 bits act as masks for the lower 16 bits. Bit 31 masks bit 15 and bit 16 masks bit 0.

1. Driver writes to GPM force wake request bit. (VV will have a Render(13_0080(15:0)) and a Media (13_00B8(15:0)) bits.)
2. The GPM responds by writing (via PLINK) to 1300B4[15:0] or 1300BC[15:0] register.
3. Driver polls (1300B4[15:0] and/or 1300BC[15:0]) status until 1... indicating that that well has completed wake sequence.

Since the registers are per thread, only the specific bit that was forced should be checked for status.
### MEDFW_REQ - Media forcewake request

<table>
<thead>
<tr>
<th>Register</th>
<th>Default Value</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FWAKEMEDIAREQ13</td>
<td>0b</td>
<td>R/W</td>
<td>Force Wake Render request 13.</td>
</tr>
<tr>
<td>FWAKEMEDIAREQ12</td>
<td>0b</td>
<td>R/W</td>
<td>Force Wake Render request 12.</td>
</tr>
<tr>
<td>FWAKEMEDIAREQ11</td>
<td>0b</td>
<td>R/W</td>
<td>Force Wake Render request 11.</td>
</tr>
<tr>
<td>FWAKEMEDIAREQ10</td>
<td>0b</td>
<td>R/W</td>
<td>Force Wake Render request 10.</td>
</tr>
<tr>
<td>FWAKEMEDIAREQ9</td>
<td>0b</td>
<td>R/W</td>
<td>Force Wake Render request 9.</td>
</tr>
<tr>
<td>FWAKEMEDIAREQ8</td>
<td>0b</td>
<td>R/W</td>
<td>Force Wake Render request 8.</td>
</tr>
<tr>
<td>FWAKEMEDIAREQ7</td>
<td>0b</td>
<td>R/W</td>
<td>Force Wake Render request 7.</td>
</tr>
<tr>
<td>FWAKEMEDIAREQ6</td>
<td>0b</td>
<td>R/W</td>
<td>Force Wake Render request 6.</td>
</tr>
<tr>
<td>FWAKEMEDIAREQ5</td>
<td>0b</td>
<td>R/W</td>
<td>Force Wake Render request 5.</td>
</tr>
</tbody>
</table>

**Default Value:** 0b

**Access:** R/W
<table>
<thead>
<tr>
<th>Field</th>
<th>Default Value</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FWAKE_MEDIAREQ4</td>
<td>0b</td>
<td>R/W</td>
<td>Force Wake Render request 4.</td>
</tr>
<tr>
<td>FWAKE_MEDIAREQ3</td>
<td>0b</td>
<td>R/W</td>
<td>Force Wake Render request 3.</td>
</tr>
<tr>
<td>FWAKE_MEDIAREQ2</td>
<td>0b</td>
<td>R/W</td>
<td>Force Wake Render request 2.</td>
</tr>
<tr>
<td>FWAKE_MEDIAREQ1</td>
<td>0b</td>
<td>R/W</td>
<td>Force Wake Render request 1.</td>
</tr>
<tr>
<td>FWAKE_MEDIAREQ0</td>
<td>0b</td>
<td>R/W</td>
<td>Force Wake Render request 0.</td>
</tr>
</tbody>
</table>
### Media Power Meter Counter

**MPMCNT - Media Power Meter Counter**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0A270h-0A273h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
<th>Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31</td>
<td>Media Power Meter Counter Overflow</td>
<td></td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td>30:0</td>
<td>Media PWRMTR Counter</td>
<td></td>
<td>RO</td>
</tr>
</tbody>
</table>
# Media Power Meter Counter No Clear

<table>
<thead>
<tr>
<th>MPMCNTCLR - Media Power Meter Counter No Clear</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 0A288h-0A28Bh</td>
</tr>
</tbody>
</table>

Formerly cleared the count and the overflow bit, but now it is just a read-only value.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31</td>
<td><strong>Media Power Meter Counter Overflow No Clear</strong></td>
<td>RO</td>
<td>Formerly cleared the overflow bit, but now it is just a read-only value.</td>
</tr>
<tr>
<td>30:0</td>
<td></td>
<td><strong>Media PWRMTR Counter No Clear</strong></td>
<td>RO</td>
<td>Formerly cleared the count, but now is just a read-only value.</td>
</tr>
</tbody>
</table>
## MEDIARC0COUNTER

<table>
<thead>
<tr>
<th>Description</th>
<th>Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>This register contains the total RC0 residency (Media powered on and clocks running) time that Media was in since boot.</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>SOXi Context Save/Restore : No</td>
<td></td>
</tr>
<tr>
<td>The 40-bit HW counter will wrap around. The only clear condition is CZ reset.</td>
<td></td>
</tr>
<tr>
<td>When 0x13_8104[15] is set to zero, bits 31:0 of the 40-bit counter are reported. When 0x13_8104[15] is set to one, bits 39:8 of the 40-bit counter are reported.</td>
<td></td>
</tr>
<tr>
<td>The units are CZ clock cycles.</td>
<td></td>
</tr>
<tr>
<td>It is up to SW to periodically read this register and do a difference from the last time it was read. The primary usage for this register is for power characterization.</td>
<td></td>
</tr>
<tr>
<td>0x13_8104[5] controls if this register should count or if it should be gated: 0= clear, 1=count</td>
<td></td>
</tr>
<tr>
<td>This register will be cleared when A024[8] is set to 1</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>MEDIARC0TIME</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Media RC0 Residency Counter.</td>
</tr>
</tbody>
</table>
**MEDIARC1COUNTER**

<table>
<thead>
<tr>
<th><strong>MEDIARC1COUNTER - MEDIARC1COUNTER</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 138114h</td>
</tr>
</tbody>
</table>

This register contains the total RC1 residency (Media powered on and clock gated) time that Media was in since boot.

SOXi Context Save/Restore : No

The 40-bit HW counter will wrap around. The only clear condition is CZ reset.

When 0x13_8104[15] is set to zero, bits 31:0 of the 40-bit counter are reported. When 0x13_8104[15] is set to one, bits 39:8 of the 40-bit counter are reported.

The units are CZ clock cycles.

It is up to SW to periodically read this register and do a difference from the last time it was read. The primary usage for this register is for power characterization.

0x13_8104[3] controls if this register should count or if it should be gated: 0= clear, 1=count

### DWord | Bit | Description |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>MEDIARC1TIME</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Media RC1 Residency Counter.</td>
</tr>
</tbody>
</table>
MEDIARC6COUNTER

<table>
<thead>
<tr>
<th>MEDIARC6COUNTER - MEDIARC6COUNTER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 13810Ch</td>
</tr>
</tbody>
</table>

This register contains the total RC6 residency (Media power gated and clock gated) time that Media was in since boot. The counter will wrap around.

SOXi Context Save/Restore: No

The time is given in units of CZ clock cycles. The counter will reset on CZ reset going high. This means that a warm reset will also reset this counter and it will also wrap around when it reaches max with no indication that an overflow occurred.

This register will freeze the count value (stop counting, but not reset) when pmu_gvd_renwakeack_nczfwoh=1.

This register will count whenever pmu_gvd_renwakeack_nczfwoh=0.

When 0x13_8104[15] is set to zero, bits 31:0 of the 40-bit counter are reported. When 0x13_8104[15] is set to one, bits 39:8 of the 40-bit counter are reported.

The units are CZ clock cycles.

It is up to SW to periodically read this register and do a difference from the last time it was read. The primary usage for this register is for power characterization.

0x13_8104[1] controls if this register should count or if it should be gated: 0= clear, 1=count

### DWord 0

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>31:0</td>
<td>MEDIARC6TIME</td>
</tr>
</tbody>
</table>

**Default Value:** 00000000h

**Access:** RO

Media Residency Counter.
## MEMBOUNDcontador

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>138120h</td>
</tr>
</tbody>
</table>

SOx Context Save/Restore : No

The 40-bit HW counter will wrap around. The only clear condition is CZ reset.

When 0x13_8104[15] is set to zero, bits 31:0 of the 40-bit counter are reported. When 0x13_8104[15] is set to one, bits 39:8 of the 40-bit counter are reported.

The units are CZ clock cycles.

It is up to SW to periodically read this register and do a difference from the last time it was read. The primary usage for this register is for power characterization.

0x13_8104[6] controls if this register should count or if it should be gated: 0= clear, 1=count

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>MEMBOUNDTIME</td>
</tr>
</tbody>
</table>

**MEMBOUNDTIME**

- Default Value: 00000000h
- Access: RO
- Render RC0 Residency Counter.
## Message Register

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000001</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>040D4h</td>
</tr>
</tbody>
</table>

### MSGREG - Message Register

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td>Mask Bits</td>
<td>0000h</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>GO_PROTOCOL_GAM_REQUEST15</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td>GO_PROTOCOL_GAM_REQUEST14</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
<td>GO_PROTOCOL_GAM_REQUEST13</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>GO_PROTOCOL_GAM_REQUEST12</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>GO_PROTOCOL_GAM_REQUEST11</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>GO_PROTOCOL_GAM_REQUEST10</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### MSGREG - Message Register

<table>
<thead>
<tr>
<th>Offset</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>GO_PROTOCOL_GAM_REQUEST9</td>
<td>0b</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>GO_PROTOCOL_GAM_REQUEST8</td>
<td>0b</td>
<td>R/W</td>
<td>Reserved</td>
</tr>
<tr>
<td>7</td>
<td>GO_PROTOCOL_GAM_REQUEST7</td>
<td>0b</td>
<td>R/W</td>
<td>Reserved</td>
</tr>
<tr>
<td>6</td>
<td>GO_PROTOCOL_GAM_REQUEST6</td>
<td>0b</td>
<td>R/W</td>
<td>Reserved</td>
</tr>
<tr>
<td>5</td>
<td>GO_PROTOCOL_GAM_REQUEST5</td>
<td>0b</td>
<td>R/W</td>
<td>Reserved</td>
</tr>
<tr>
<td>4</td>
<td>GO_PROTOCOL_GAM_REQUEST4</td>
<td>0b</td>
<td>R/W</td>
<td>Reserved</td>
</tr>
<tr>
<td>3</td>
<td>GO_PROTOCOL_GAM_REQUEST3</td>
<td>0b</td>
<td>R/W</td>
<td>Reserved</td>
</tr>
<tr>
<td>2</td>
<td>GO_PROTOCOL_GAM_REQUEST2</td>
<td>0b</td>
<td>R/W</td>
<td>Reserved</td>
</tr>
<tr>
<td>1</td>
<td>GO_PROTOCOL_GAM_REQUEST1</td>
<td>0b</td>
<td>R/W</td>
<td>Reserved</td>
</tr>
<tr>
<td>0</td>
<td>GO_PROTOCOL_GAM_REQUEST0</td>
<td>1b</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### MSGREG - Message Register

<table>
<thead>
<tr>
<th>Access</th>
<th>R/W</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>0 - GPM to GAM Busy Ack Indication.</td>
<td></td>
</tr>
<tr>
<td>1 - GPM to GAM Idle Ack Indication.</td>
<td></td>
</tr>
</tbody>
</table>
Messaging Register for GPMunit

MSG_GPM - Messaging Register for GPMunit

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>00C00h</td>
</tr>
</tbody>
</table>

Message registers have bit-wise masking applied for writes. The register consists of 16 bits of data in [15:0], and 16 bits of corresponding masks in [31:16]. To set bit0, for example, the data would be 0x0001_0001. To clear bit0, for example, the data would be 0x0001_0000. Note that mask bit is the data bit offset + 16. Message registers are protected from non-GT writes via the Message Channel.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td><strong>GPM Messages Bit 15</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Placeholder for GPM Messages. RPMunit could self-clear these bits upon sampling.</td>
</tr>
<tr>
<td>14</td>
<td></td>
<td><strong>GPM Messages Bit 14</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Placeholder for GPM Messages. RPMunit could self-clear these bits upon sampling.</td>
</tr>
<tr>
<td>13</td>
<td></td>
<td><strong>GPM Messages Bit 13</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Placeholder for GPM Messages. RPMunit could self-clear these bits upon sampling.</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td><strong>GPM Messages Bit 12</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Placeholder for GPM Messages. RPMunit could self-clear these bits upon sampling.</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td><strong>GPM Messages Bit 11</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Placeholder for GPM Messages. RPMunit could self-clear these bits upon sampling.</td>
</tr>
<tr>
<td>MSG_GPM - Messaging Register for GPMunit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10 GPM Messages Bit 10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Access: R/W</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Placeholder for GPM Messages.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RPMunit could self-clear these bits upon sampling.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9  GPM Messages Bit 9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Access: R/W</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Placeholder for GPM Messages.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RPMunit could self-clear these bits upon sampling.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8  GPM Messages Bit 8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Access: R/W</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Placeholder for GPM Messages.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RPMunit could self-clear these bits upon sampling.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7  GPM Messages Bit 7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Access: R/W</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Placeholder for GPM Messages.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RPMunit could self-clear these bits upon sampling.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6  GPM Messages Bit 6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Access: R/W</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Placeholder for GPM Messages.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RPMunit could self-clear these bits upon sampling.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5  GPM Messages Bit 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Access: R/W</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Placeholder for GPM Messages.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RPMunit could self-clear these bits upon sampling.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4  Request to send CPD Exit Ack Message on EventBus (U2C)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Access: R/W</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Request from GPMunit for RPMunit to send CPD_EXIT_ACK message on the Eventbus. RPMunit self-clears this bit upon sampling.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3  Request to send CPD Enter Ack Message on EventBus (U2C)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Access: R/W</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Request from GPMunit for RPMunit to send CPD_ENTER_ACK message on the Eventbus. RPMunit self-clears this bit upon sampling.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### MSG_GPM - Messaging Register for GPMunit

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Request to send Credit Active Deassert Message on EventBus (U2C)</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>Request from GPMunit for RPMunit to send CREDIT_ACTIVE_DEASSERT message on the Eventbus. RPMunit self-clears this bit upon sampling.</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Request to send Credit Active Assert Message on EventBus (U2C)</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>Request from GPMunit for RPMunit to send CREDIT_ACTIVE_ASSERT message on the Eventbus. RPMunit self-clears this bit upon sampling.</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>Request to send IDI Shutdown Ack Message on EventBus (U2C)</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>Request from GPMunit for RPMunit to send IDI_SHUTDOWN_ACK message on the Eventbus. RPMunit self-clears this bit upon sampling.</td>
<td></td>
</tr>
</tbody>
</table>
### Messaging Register for MDRBunit

<table>
<thead>
<tr>
<th>MSG_MDRB - Messaging Register for MDRBunit</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000001</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 00C08h</td>
</tr>
</tbody>
</table>

Message registers have bit-wise masking applied for writes. The register consists of 16 bits of data in [15:0], and 16 bits of corresponding masks in [31:16].

To set bit0, for example, the data would be 0x0001_0001.

To clear bit0, for example, the data would be 0x0001_0000.

Note that mask bit is the data bit offset + 16. Message registers are protected from non-GT writes via the Message Channel.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>RO</td>
</tr>
<tr>
<td>15:2</td>
<td>MDRB Messages</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>1</td>
<td>RFO Enable/Disable Ack for RPM (internal) RFO Request</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>RFO Enable/Disable Ack for Internal RFO Request.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Enable Ack = 1'b1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Disable Ack = 1'b0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>RFO Enable/Disable Ack for U2C (Eventbus) RFO Request</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default Value:</td>
<td>1b</td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>RFO Enable/Disable Ack for U2C RFO Request.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Enable Ack = 1'b1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Disable Ack = 1'b0</td>
<td></td>
</tr>
</tbody>
</table>
### Messaging Register for MGSRunit

<table>
<thead>
<tr>
<th>MSG_MGSR - Messaging Register for MGSRunit</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 00C04h</td>
</tr>
</tbody>
</table>

Message registers have bit-wise masking applied for writes. The register consists of 16 bits of data in [15:0], and 16 bits of corresponding masks in [31:16].

To set bit0, for example, the data would be 0x0001_0001.

To clear bit0, for example, the data would be 0x0001_0000.

Note that mask bit is the data bit offset + 16. Message registers are protected from non-GT writes via the Message Channel.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td>15:0</td>
<td>MGSR Messages</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

Placeholder for MGSR Messages.

MGSRunit could self-clear these bits upon sampling.
**MFC_AVC_CABAC_INSERTION_COUNT**

<table>
<thead>
<tr>
<th><strong>AVC_CABAC_INSERTION_COUNT</strong> - <strong>MFC_AVC_CABAC_INSERTION_COUNT</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space:    MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project:    CHV, BSW</td>
</tr>
<tr>
<td>Source:    VideoCS</td>
</tr>
<tr>
<td>Default Value:    0x00000000</td>
</tr>
<tr>
<td>Access:    RO</td>
</tr>
<tr>
<td>Size (in bits):    32</td>
</tr>
<tr>
<td>Trusted Type:    1</td>
</tr>
<tr>
<td>Address:    128ACh</td>
</tr>
<tr>
<td>Valid Projects:    CHV, BSW</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>MFC AVC Cabac Insertion Count</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Total number of bytes in the bitstream output before for the CABAC zero word insertion. This count is updated each time when the insertion count is incremented.</td>
</tr>
</tbody>
</table>
## MFC_VIN_AVD_ERROR_CNTR - MFC_AVC Bitstream Decoding Front-End Parsing Logic Error Counter

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
</tbody>
</table>

### Register Details
- **Register Space:** MMIO: 0/2/0
- **Project:** CHV, BSW
- **Source:** VideoCS
- **Default Value:** 0x00000000
- **Access:** R/W
- **Size (in bits):** 32
- **Address:** 12804h
- **Valid Projects:** CHV, BSW
- **Trusted Type:** 1

The **MFC_AVC Bitstream Decoding Front-End Parsing Logic Error Counter** is used to track errors during the decoding process. The register allows monitoring and debugging issues related to the parsing logic, ensuring the integrity of the decoded bitstream.
# MFC Image Status Control

<table>
<thead>
<tr>
<th><strong>MFC_IMAGE_STATUS_CONTROL - MFC Image Status Control</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> VideoCS</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Trusted Type:</strong> 1</td>
</tr>
<tr>
<td><strong>Address:</strong> 128B8h</td>
</tr>
<tr>
<td><strong>Valid Projects:</strong> CHV, BSW</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>Cumulative slice delta QP</strong></td>
</tr>
<tr>
<td></td>
<td>23:16</td>
<td><strong>QP Value</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>suggested slice QP delta value for frame level Rate control. This value can be +ve or -ve</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td><strong>QP-Polarity Change</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cumulative slice delta QP polarity change.</td>
</tr>
<tr>
<td>14:13</td>
<td></td>
<td><strong>Num-Pass Polarity Change</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Number of passes after cumulative slice delta QP polarity changes.</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
<tr>
<td>11:8</td>
<td></td>
<td><strong>Total Num-Pass</strong></td>
</tr>
<tr>
<td>7:4</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td><strong>Missing Huffman Code</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Jpeg HW encoder reports if Huffman table entry is missing.</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td><strong>Panic</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Panic triggered to avoid too big packed file.</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td><strong>Frame Bit Count</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Frame Bit count over-run/under-run flag</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td><strong>Max Conformance Flag</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Max Macroblock conformance flag or Frame Bit count over-run/under-run</td>
</tr>
</tbody>
</table>
MFC Image Status Mask

<table>
<thead>
<tr>
<th>MFC_IMAGE_STATUS_MASK - MFC Image Status Mask</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: VideoCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Access: RO</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Trusted Type: 1</td>
</tr>
<tr>
<td>Address: 128B4h</td>
</tr>
<tr>
<td>Valid Projects: CHV, BSW</td>
</tr>
</tbody>
</table>

This register stores the image status(flags).

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>Control Mask</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Control Mask for dynamic frame repeat.</td>
</tr>
</tbody>
</table>
## MFC QP Status Count

<table>
<thead>
<tr>
<th><strong>MFC_QUP_CT - MFC QP Status Count</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> VideoCS</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Trusted Type:</strong> 1</td>
</tr>
<tr>
<td><strong>Address:</strong> 128BCh</td>
</tr>
<tr>
<td><strong>Valid Projects:</strong> CHV, BSW</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>Cumulative QP Adjust</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format:</strong> U8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cumulative QP adjustment after multiple passes. If there is no need to multi-pass, this value would be zero. (This is in sign magnitude form).</td>
</tr>
<tr>
<td>0</td>
<td>23:0</td>
<td><strong>Cumulative QP</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format:</strong> U24</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cumulative QP for all MB of a Frame (Can be used for computing average QP).</td>
</tr>
</tbody>
</table>
# MFD Error Status

<table>
<thead>
<tr>
<th>MFD_ERROR_STATUS - MFD Error Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: VideoCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Access: RO</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Trusted Type: 1</td>
</tr>
<tr>
<td>Address: 12800h</td>
</tr>
<tr>
<td>Valid Projects: CHV, BSW</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:20</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This field is currently reserved</td>
</tr>
<tr>
<td>19:16</td>
<td></td>
<td><strong>AVC Short Format Error Flags</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Exists If: // AVC Short Format == True</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit-stream error detected by VLD short format bit-stream decoder. These flags are reset at the beginning of a frame and updated until starting of another frame.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[19] – Slice Type SE Error Flag – Invalid Slice Type SE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[18] – MMCO SE Error Flag – Invalid memory management control operation SE. MMCO Loop does not end (mmco control != 0) even after all MMCO SEs are decoded OR MMCO SEs are still being decoded and MMCO SE loop end (mmco control == 0) is hit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[17] – Reordering IDC Error Flag – Syntax Element modification_of_pic_nums_idc &gt;= 6 OR modification_of_pic_nums_idc != 3 (end of reordering loop) but reordering count has already hit maximum value</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[16] – Premature bitstream end is hit before finishing slice header decode</td>
</tr>
<tr>
<td></td>
<td>15:0</td>
<td><strong>Bit-stream Error flags</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Exists If: // AVC CAVLC, AVC CABAC, VC1 and MPEG2 == True</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bitstream error detected by the VLD bitstream decoder. These flags are reset at the beginning of a frame and updated until starting of another frame.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AVC CAVLC: Please refer to AVC CAVLC table for each bit field</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AVC CABAC: Please refer to AVC CABAC table for each bit field</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VC1: Please refer to VC1 table for each bit field</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MPEG2: Please refer to MPEG2 table for each bit field</td>
</tr>
</tbody>
</table>
# MFD Picture Parameter

<table>
<thead>
<tr>
<th>MFD_PICTURE_PARAM - MFD Picture Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> VideoCS</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Trusted Type:</strong> 1</td>
</tr>
<tr>
<td><strong>Address:</strong> 12820h</td>
</tr>
<tr>
<td><strong>Valid Projects:</strong> CHV, BSW</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Reserved</strong></td>
</tr>
</tbody>
</table>

- **Format:** MBZ
## MFX_Memory_Latency_Count1

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Max Request Count</strong></td>
<td>This field indicates the maximum number of requests allowed by the memory sub-system channel.</td>
</tr>
<tr>
<td><strong>Current Request Count</strong></td>
<td>This field indicates the number of requests currently outstanding in the memory sub-system. This field should report with a value of zero at the end of frame; otherwise the motion compensation engine is most likely hung waiting for read data to be returned from sub-system.</td>
</tr>
<tr>
<td><strong>MFX Reference picture read request - Max Latency Count in 8xMedia clock cycles</strong></td>
<td>This field reports the maximum memory latency count on all reference reads requested by the motion compensation engine.</td>
</tr>
<tr>
<td><strong>MFX Reference picture read request - Min Latency Count in 8xMedia clock cycles</strong></td>
<td>This field reports the minimum memory latency count on all reference reads requested by the motion compensation engine.</td>
</tr>
</tbody>
</table>
# MFX0 Context Element Descriptor (High Part)

<table>
<thead>
<tr>
<th>MFX0_CTX_EDR_H - MFX0 Context Element Descriptor (High Part)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 04444h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>MFX0 Context Element Descriptor (High Part)</td>
<td>00000000h</td>
<td>R/W</td>
</tr>
</tbody>
</table>
# MFX0 Context Element Descriptor (Low Part)

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>MFX0 Context Element Descriptor (Low Part)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000009h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

- **Register Space:** MMIO: 0/2/0
- **Project:** CHV, BSW
- **Source:** PRM
- **Default Value:** 0x00000009
- **Size (in bits):** 32
- **Address:** 04440h
### MFX0 Context Element Descriptor (Low Part)

<table>
<thead>
<tr>
<th><strong>MFX0_CTX_EDR_L - MFX0 Context Element Descriptor (Low Part)</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000009</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 04440h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>MFX0 Context Element Descriptor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000009h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

**MFX0 Context Element Descriptor**

- Default Value: 00000009h
- Access: R/W
# MFX0 PDP0/PML4/PASID Descriptor (High Part)

<table>
<thead>
<tr>
<th><strong>MFX0_CTX_PDP0_H</strong> - MFX0 PDP0/PML4/PASID Descriptor (High Part)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 0444Ch</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>DWord</strong></th>
<th><strong>Bit</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31-0</td>
<td><strong>MFX0 PDP0/PML4/PASID Descriptor (High Part)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>
# MFX0 PDP0/PML4/PASID Descriptor (Low Part)

**MFX0_CTX_PDP0_L - MFX0 PDP0/PML4/PASID Descriptor (Low Part)**

<table>
<thead>
<tr>
<th>Register Space</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits)</td>
<td>32</td>
</tr>
<tr>
<td>Address</td>
<td>04448h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>MFX0 PDP0/PML4/PASID Descriptor (Low Part)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

---

**Doc Ref # IHD-OS-CHV-BSW-Vol 2c- 10.15**
## MFX0 PDP1 Descriptor Register (High Part)

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space:</td>
<td>MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>04454h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>MFX0 PDP1 Descriptor Register (High Part)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>
### MFX0 PDP1 Descriptor Register (Low Part)

<table>
<thead>
<tr>
<th>MFX0_CTX_PDP1_L</th>
<th>MFX0 PDP1 Descriptor Register (Low Part)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space:</td>
<td>MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>04450h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>MFX0 PDP1 Descriptor Register (Low Part)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>
# MFX0 PDP2 Descriptor Register (High Part)

<table>
<thead>
<tr>
<th>MFX0_CTX_PDP2_H - MFX0 PDP2 Descriptor Register (High Part)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 0445Ch</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>MFX0 PDP2 Descriptor Register (High Part)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>
### MFX0 PDP2 Descriptor Register (Low Part)

<table>
<thead>
<tr>
<th>MFX0_CTX_PDP2_L - MFX0 PDP2 Descriptor Register (Low Part)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 04458h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>MFX0 PDP2 Descriptor Register (Low Part)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>
# MFX0 PDP3 Descriptor Register (High Part)

<table>
<thead>
<tr>
<th>MFX0_CTX_PDP3_H - MFX0 PDP3 Descriptor Register (High Part)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space:</td>
</tr>
<tr>
<td>Project:</td>
</tr>
<tr>
<td>Source:</td>
</tr>
<tr>
<td>Default Value:</td>
</tr>
<tr>
<td>Size (in bits):</td>
</tr>
<tr>
<td>Address:</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>MFX0 PDP3 Descriptor Register (High Part)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>R/W</td>
</tr>
</tbody>
</table>
# MFX0 PDP3 Descriptor Register (Low Part)

<table>
<thead>
<tr>
<th>MFX0_CTX_PDP3_L - MFX0 PDP3 Descriptor Register (Low Part)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 04460h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>MFX0 PDP3 Descriptor Register (Low Part)</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
</tbody>
</table>
# MFX1 Context Element Descriptor (High Part)

## MFX1_CTX_EDR_H - MFX1 Context Element Descriptor (High Part)

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>04484h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>MFX1 Context Element Descriptor (High Part)</strong></td>
<td>00000000h</td>
<td>R/W</td>
</tr>
</tbody>
</table>
## MFX1 Context Element Descriptor (Low Part)

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>MFX1 Context Element Descriptor (Low Part)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000009h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

Register Space: MMIO: 0/2/0  
Project: CHV, BSW  
Source: PRM  
Default Value: 0x00000009  
Size (in bits): 32  
Address: 04480h
### MFX1 Context Element Descriptor (Low Part)

<table>
<thead>
<tr>
<th><strong>MFX1_CTX_EDR_L - MFX1 Context Element Descriptor (Low Part)</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000009</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 04480h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>MFX1 Context Element Descriptor</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000009h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>
# MFX1 PDP0/PML4/PASID Descriptor (High Part)

<table>
<thead>
<tr>
<th>MFX1_CTX_PDP0_H - MFX1 PDP0/PML4/PASID Descriptor (High Part)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 0448Ch</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>MFX1 PDP0/PML4/PASID Descriptor (High Part)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

## MFX1 PDP0/PML4/PASID Descriptor (Low Part)

<table>
<thead>
<tr>
<th>MFX1_CTX_PDP0_L - MFX1 PDP0/PML4/PASID Descriptor (Low Part)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 04488h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>MFX1 PDP0/PML4/PASID Descriptor (Low Part)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
</tbody>
</table>


# MFX1 PDP1 Descriptor Register (High Part)

<table>
<thead>
<tr>
<th>MFX1_CTX_PDP1_H - MFX1 PDP1 Descriptor Register (High Part)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 04494h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>MFX1 PDP1 Descriptor Register (High Part)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>
# MFX1 PDP1 Descriptor Register (Low Part)

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>MFX1 PDP1 Descriptor Register (Low Part)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0x00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>
### MFX1 PDP2 Descriptor Register (High Part)

<table>
<thead>
<tr>
<th>register</th>
<th>Address</th>
<th>Description</th>
<th>DWord</th>
<th>Bit</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>MFX1_CTX_PDP2_H</td>
<td>0449Ch</td>
<td>MFX1 PDP2 Descriptor Register (High Part)</td>
<td>0</td>
<td>31:0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0x00000000h</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Register Space:** MMIO: 0/2/0  
**Project:** CHV, BSW  
**Source:** PRM  
**Default Value:** 0x00000000  
**Size (in bits):** 32
## MFX1 PDP2 Descriptor Register (Low Part)

<table>
<thead>
<tr>
<th><strong>MFX1_CTX_PDP2_L - MFX1 PDP2 Descriptor Register (Low Part)</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 04498h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>DWORD</strong></th>
<th><strong>Bit</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>MFX1 PDP2 Descriptor Register (Low Part)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>
# MFX1 PDP3 Descriptor Register (High Part)

<table>
<thead>
<tr>
<th>MFX1_CTX_PDP3_H - MFX1 PDP3 Descriptor Register (High Part)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 044A4h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>MFX1 PDP3 Descriptor Register (High Part)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>
## MFX1 PDP3 Descriptor Register (Low Part)

<table>
<thead>
<tr>
<th>MFX1CTX_PDP3_L - MFX1 PDP3 Descriptor Register (Low Part)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 044A0h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>MFX1 PDP3 Descriptor Register (Low Part)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>
# MFX Frame BitStream SE/BIN Count

<table>
<thead>
<tr>
<th>MFX_SE-BIN_CT - MFX Frame BitStream SE/BIN Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: VideoCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Access: RO</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Trusted Type: 1</td>
</tr>
<tr>
<td>Address: 1286Ch</td>
</tr>
<tr>
<td>Valid Projects: CHV, BSW</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>MFX Frame Bit-stream SE/BIN Count</td>
</tr>
</tbody>
</table>
|       |     | Total number of BINs/SEs decoded in current frame. This number is used with frame performance count to derive Bin/clk or SE/clk.
## MFX Frame Macroblock Count

### MFX_MB_COUNT - MFX Frame Macroblock Count

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>VideoCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>RO</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Trusted Type:</td>
<td>1</td>
</tr>
<tr>
<td>Address:</td>
<td>12868h</td>
</tr>
<tr>
<td>Valid Projects:</td>
<td>CHV, BSW</td>
</tr>
</tbody>
</table>

This register stores the number of Macro-blocks decoded/encoded in current frame. This register is not part of hardware context save and restore.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:20</td>
<td>MBZ</td>
</tr>
<tr>
<td></td>
<td>Exists If:</td>
<td>// JPEG == True</td>
</tr>
<tr>
<td></td>
<td>Format:</td>
<td>MBZ</td>
</tr>
<tr>
<td></td>
<td>This field is currently reserved</td>
<td></td>
</tr>
<tr>
<td></td>
<td>31:16</td>
<td>Intra MB Count</td>
</tr>
<tr>
<td></td>
<td>Exists If:</td>
<td>// AVC CAVLC, AVC CABAC, VC1 and MPEG2 == True</td>
</tr>
<tr>
<td></td>
<td>Format:</td>
<td>U16</td>
</tr>
<tr>
<td></td>
<td>19:0</td>
<td>JPEG Block Count</td>
</tr>
<tr>
<td></td>
<td>Exists If:</td>
<td>// JPEG == True</td>
</tr>
<tr>
<td></td>
<td>Format:</td>
<td>U20</td>
</tr>
<tr>
<td></td>
<td>This 20-bit field indicates the number of 8x8 blocks within the JPEG frame. This field is clear at the start of decoding a new frame.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>15:0</td>
<td>Number of MB Concealment</td>
</tr>
<tr>
<td></td>
<td>Exists If:</td>
<td>// AVC CAVLC, AVC CABAC, VC1 and MPEG2 == True</td>
</tr>
<tr>
<td></td>
<td>This 16-bit field indicates the number of MB is concealed by hardware. This field is clear at the start of decoding a new frame.</td>
<td></td>
</tr>
</tbody>
</table>
# MFX Frame Motion Comp Miss Count

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>VideoCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>RO</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Trusted Type:</td>
<td>1</td>
</tr>
</tbody>
</table>

Address: 12888h
Valid Projects: CHV, BSW

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

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

<table>
<thead>
<tr>
<th>MFX_READ_CT - MFX Frame Motion Comp Read Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: VideoCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Access: RO</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Trusted Type: 1</td>
</tr>
<tr>
<td>Address: 12484h</td>
</tr>
</tbody>
</table>

This register stores the total number of reference picture read requests made by the Motion Compensation engine per frame. This register is not part of hardware context save and restore.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:20</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format:</strong> MBZ</td>
</tr>
<tr>
<td>19:0</td>
<td></td>
<td><strong>MFX Frame Motion Comp CL read request Count</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Total number of reference picture read requests by the motion compensation engine per frame.</td>
</tr>
</tbody>
</table>

---

*Doc Ref # IHD-OS-CHV-BSW-Vol 2c-10.15*
MFX Frame Row-Store/BitStream Read Count

<table>
<thead>
<tr>
<th>MFX_ROW_PER_BS_COUNT - MFX Frame Row-Store/BitStream Read Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: VideoCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Access: RO</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Trusted Type: 1</td>
</tr>
<tr>
<td>Address: 12880h</td>
</tr>
<tr>
<td>Valid Projects: CHV, BSW</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td>Format:</td>
<td>MBZ</td>
</tr>
<tr>
<td>15:0</td>
<td></td>
<td>MFX row-stored/bit-stream read request Count</td>
</tr>
<tr>
<td></td>
<td>Total number of row-stored/bit-stream read requests sent by the memory pre-fetch engine per frame.</td>
<td></td>
</tr>
</tbody>
</table>
## MFX LRA 0

### MFX_LRA_0 - MFX LRA 0

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>MFX LRA1 Max</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 01111111b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Maximum value of programmable LRA1.</td>
</tr>
<tr>
<td>23:16</td>
<td></td>
<td><strong>MFX LRA1 Min</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 01000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td>15:8</td>
<td></td>
<td>Minimum value of programmable LRA1.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>MFX LRA0 Max</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00111111b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Maximum value of programmable LRA0.</td>
</tr>
<tr>
<td>7:0</td>
<td></td>
<td><strong>MFX LRA0 Min</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Minimum value of programmable LRA0.</td>
</tr>
</tbody>
</table>
## MFX LRA 1

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>MFX LRA3 Max</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 11111111b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Maximum value of programmable LRA3.</td>
</tr>
<tr>
<td>23:16</td>
<td></td>
<td><strong>MFX LRA3 Min</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 11000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Minimum value of programmable LRA3.</td>
</tr>
<tr>
<td>15:8</td>
<td></td>
<td><strong>MFX LRA2 Max</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 10111111b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Maximum value of programmable LRA2.</td>
</tr>
<tr>
<td>7:0</td>
<td></td>
<td><strong>MFX LRA2 Min</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 10000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Minimum value of programmable LRA2.</td>
</tr>
</tbody>
</table>
## MFX_LRA_2 - MFX LRA 2

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:8</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td>7:6</td>
<td><strong>VCS LRA</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td>5:4</td>
<td><strong>VMX LRA</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 01b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td>3:2</td>
<td><strong>VMC LRA</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 10b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td>1:0</td>
<td><strong>VCR LRA</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 11b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

- **Register Space:** MMIO: 0/2/0
- **Project:** CHV, BSW
- **Source:** PRM
- **Default Value:** 0x0000001B
- **Size (in bits):** 32

**Address:** 04A58h

---

### Reserved

**Default Value:** 000000h

**Access:** RO

**Description:** Reserved

---

### VCS LRA

**Default Value:** 00b

**Access:** R/W

**Description:** Which LRA should VCS use.

---

### VMX LRA

**Default Value:** 01b

**Access:** R/W

**Description:** Which LRA should VMX use.

---

### VMC LRA

**Default Value:** 10b

**Access:** R/W

**Description:** Which LRA should VMC use.

---

### VCR LRA

**Default Value:** 11b

**Access:** R/W

**Description:** Which LRA should VCRSL1 use.
# MFX LRA SL1 0

## MFX_LRA_SL1_0 - MFX LRA SL1 0

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td>MFX SL1 LRA1 Max</td>
<td>01111111b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Maximum value of programmable LRA1.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>23:16</td>
<td>MFX SL1 LRA1 Min</td>
<td>01000000b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Minimum value of programmable LRA1.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>15:8</td>
<td>MFX SL1 LRA0 Max</td>
<td>00111111b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Maximum value of programmable LRA0.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>7:0</td>
<td>MFX SL1 LRA0 Min</td>
<td>00000000b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Minimum value of programmable LRA0.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## MFX_LRA_SL1_1 - MFX LRA SL1 1

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0xFFC0BF80</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>04A64h</td>
</tr>
</tbody>
</table>

### DWord 0

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>31:24</td>
<td><strong>MFX SL1 LRA3 Max</strong></td>
<td>11111111b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td><strong>MFX SL1 LRA3 Min</strong></td>
<td>11000000b</td>
<td>R/W</td>
</tr>
</tbody>
</table>

Maximum value of programmable LRA3.

### DWord 23:16

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>23:16</td>
<td><strong>MFX SL1 LRA2 Max</strong></td>
<td>10111111b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td><strong>MFX SL1 LRA2 Min</strong></td>
<td>10000000b</td>
<td>R/W</td>
</tr>
</tbody>
</table>

Maximum value of programmable LRA2.

### DWord 15:8

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
</table>

### DWord 7:0

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
</table>

Minimum value of programmable LRA2.
### MFX LRA SL1 2

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x0000001B</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>04A68h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:8</td>
<td>Reserved</td>
<td>000000h</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R/W</td>
</tr>
<tr>
<td>7:6</td>
<td></td>
<td>VCSSL1 LRA</td>
<td>00b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5:4</td>
<td></td>
<td>VMXSL1 LRA</td>
<td>01b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3:2</td>
<td></td>
<td>VMCSL1 LRA</td>
<td>10b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1:0</td>
<td></td>
<td>VCRSL1 LRA</td>
<td>11b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Which LRA should VCSSL1 use.

Which LRA should VMXSL1 use.

Which LRA should VMCSL1 use.

Which LRA should VCRSL1 use.
MFX Memory Latency Count2

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:26</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
<tr>
<td>25:0</td>
<td></td>
<td>MFX Reference picture read request - Accumulative Memory Latency Count for the entire frame in 8xMedia clock cycles</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The accumulative memory latency count of all reference reads requested by motion compensative engine per frame. This number is used with MFX Frame Motion Comp Read Count to derive average memory latency.</td>
</tr>
</tbody>
</table>
MFX Memory Latency Count3

<table>
<thead>
<tr>
<th>MFX_LAT_CT3 - MFX Memory Latency Count3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: VideoCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Access: RO</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Trusted Type: 1</td>
</tr>
<tr>
<td>Address: 12878h</td>
</tr>
<tr>
<td>Valid Projects: CHV, BSW</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
</table>
| 0     | 31:24 | **Max Request Count**  
This field indicates the maximum number of requests allowed by the memory sub-system channel. |
|       | 23:16 | **Current Request Count**  
This field indicates the number of requests currently outstanding in the memory sub-system. This field should report with a value of zero at the end of frame; otherwise the pre-fetch engine most likely hung waiting for read data to be returned from sub-system. |
|       | 15:8  | **MFX row-stored/bit-stream read request - Max Latency Count in 8xMedia clock cycles**  
This field reports the maximum memory latency count on all row-stored/bit-stream reads requested by the memory pre-fetch engine. |
|       | 7:0   | **MFX row-stored/bit-stream read request - Min Latency Count in 8xMedia clock cycles**  
This field reports the minimum memory latency count on all row-stored/bit-stream reads requested by the memory pre-fetch engine. |
MFX Memory Latency Count4

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

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

Register Space: MMIO: 0/2/0
Project: CHV, BSW
Source: VideoCS
Default Value: 0x00000000
Access: RO
Size (in bits): 32
Trusted Type: 1
Address: 1287Ch
Valid Projects: CHV, BSW
# MFX Pipeline Status Flags

<table>
<thead>
<tr>
<th>MFX_STATUS_FLAGS - MFX Pipeline Status Flags</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> VideoCS</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Trusted Type:</strong> 1</td>
</tr>
<tr>
<td><strong>Address:</strong> 12838h</td>
</tr>
<tr>
<td><strong>Valid Projects:</strong> CHV, BSW</td>
</tr>
</tbody>
</table>

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

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

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>AVC/VC1 Short Format Mode is in use</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>AVC/VC1 Long Format Mode is in use</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0b</td>
<td>Not in Stitch Mode</td>
<td></td>
</tr>
<tr>
<td>1b</td>
<td>In the Special Stitch Mode</td>
<td></td>
</tr>
</tbody>
</table>
# MFX Slice Performance Count

<table>
<thead>
<tr>
<th>MFX_SLICE_PERFORM_CT - MFX Slice Performance Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: VideoCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Access: RO</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Trusted Type: 1</td>
</tr>
<tr>
<td>Address: 12864h</td>
</tr>
<tr>
<td>Valid Projects: CHV, BSW</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>MFX Frame Performance Count</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Total number of clocks between slice start and slice end. This count is incremented on crm_clk</td>
</tr>
</tbody>
</table>
### MGGC - MGGC

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000028</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>108040h</td>
</tr>
</tbody>
</table>

Mirror of GMCH Graphics Control Register. Simply a read-only alias to the configuration registers. Used by the MBC for boot context fetches.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:15</td>
<td>RESERVED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>VAMEN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
</tbody>
</table>

Enables the use of the iGFX engines for Versatile Acceleration.
1 - iGFX engines are in Versatile Acceleration Mode. Device 2 Class Code is 048000h.
0 - iGFX engines are in iGFX Mode. Device 2 Class Code is 030000h.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>13:10</td>
<td>RESERVED</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>9:8</td>
<td>GGMS</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
</tbody>
</table>

GTT Graphics Memory Size (GGMS):
This field is used to select the amount of Main Memory that is pre-allocated to support the Internal Graphics Translation Table. The BIOS ensures that memory is pre-allocated only when Internal graphics is enabled.
GSM is assumed to be a contiguous physical DRAM space with DSM, and BIOS needs to allocate a contiguous memory chunk. Hardware will derive the base of GSM from DSM only using the GSM size programmed in the register.
Hardware functionality in case of programming this value to Reserved is not guaranteed.
0h: No Preallocated Memory
1h: 2MB of Preallocated Memory
2h: 4MB of Preallocated Memory
3h: 8MB of Preallocated Memory
### Graphics Mode Select (GMS)

This field is used to select the amount of Main Memory that is pre-allocated to support the Internal Graphics device in VGA (non-linear) and Native (linear) modes. The BIOS ensures that memory is pre-allocated only when Internal graphics is enabled.

Hardware does not clear or set any of these bits automatically based on IGD being disabled/enabled. BIOS Requirement: BIOS must not set this field to 0h if IVD (bit 1 of this register) is 0.

<table>
<thead>
<tr>
<th>Value</th>
<th>Memory Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>0MB</td>
</tr>
<tr>
<td>1h</td>
<td>32MB</td>
</tr>
<tr>
<td>2h</td>
<td>64MB</td>
</tr>
<tr>
<td>3h</td>
<td>96MB</td>
</tr>
<tr>
<td>4h</td>
<td>128MB</td>
</tr>
<tr>
<td>5h</td>
<td>160MB</td>
</tr>
<tr>
<td>6h</td>
<td>192MB</td>
</tr>
<tr>
<td>7h</td>
<td>224MB</td>
</tr>
<tr>
<td>8h</td>
<td>256MB</td>
</tr>
<tr>
<td>9h</td>
<td>288MB</td>
</tr>
<tr>
<td>Ah</td>
<td>320MB</td>
</tr>
<tr>
<td>8h</td>
<td>352MB</td>
</tr>
<tr>
<td>Ch</td>
<td>384MB</td>
</tr>
<tr>
<td>Dh</td>
<td>416MB</td>
</tr>
<tr>
<td>Eh</td>
<td>448MB</td>
</tr>
<tr>
<td>Fh</td>
<td>480MB</td>
</tr>
<tr>
<td>10h</td>
<td>512MB</td>
</tr>
<tr>
<td>11h</td>
<td>528MB</td>
</tr>
<tr>
<td>12h</td>
<td>552MB</td>
</tr>
<tr>
<td>13h</td>
<td>576MB</td>
</tr>
<tr>
<td>14h</td>
<td>600MB</td>
</tr>
<tr>
<td>15h</td>
<td>624MB</td>
</tr>
<tr>
<td>16h</td>
<td>648MB</td>
</tr>
<tr>
<td>17h</td>
<td>672MB</td>
</tr>
<tr>
<td>18h</td>
<td>704MB</td>
</tr>
<tr>
<td>19h</td>
<td>728MB</td>
</tr>
<tr>
<td>1Ah</td>
<td>752MB</td>
</tr>
<tr>
<td>18h</td>
<td>776MB</td>
</tr>
<tr>
<td>1Ch</td>
<td>800MB</td>
</tr>
<tr>
<td>1Dh</td>
<td>824MB</td>
</tr>
<tr>
<td>1Eh</td>
<td>Reserved</td>
</tr>
<tr>
<td>1Fh</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>20h</td>
<td>1024MB (Not supported for CHV, BSW)</td>
</tr>
</tbody>
</table>
### MGGC - MGCC

<table>
<thead>
<tr>
<th>Address</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>30h:1536MB</td>
<td>(Not supported for CHV, BSW)</td>
</tr>
<tr>
<td>40h:2048MB</td>
<td>(Not supported for CHV, BSW)</td>
</tr>
<tr>
<td>80h:4096MB</td>
<td>(Not supported for CHV, BSW)</td>
</tr>
<tr>
<td>81h - FF:</td>
<td>Reserved</td>
</tr>
<tr>
<td>Other =</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

When GMS != '0 (and VD=0):

Address[31:0] is compared with VGA memory range. (The VGA memory range is A_0000h to B_FFFFh.). If there is a match and MSE = 1 and MEMRD or MEMWR, the access will route as a Rmdwgamemen_cr cycle on the RMbus. If the RMbus returns a hit the GVD will select the command. As well, when 0 the GVD will check if sldown3_address[15:0] is one of the VGA IO register range. (The VGA IO range is 03B0h - 03BBh and 03C0h - 03DFh.) If there is a match and IOSE = 1 and the SCL command is either an IORD or IOWR, the GVD will initiate a (VGA) register cycle on the RMbus. If the RMbus returns a hit the GVD will select the command.

When GMS == '0 :

No address compare will occur against VGA memory range or the VGA IO register range. Also, CC[15:8] is changed to 8'h80 from 8'h00.

<table>
<thead>
<tr>
<th>REG</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>RESERVED</td>
<td>0b</td>
</tr>
<tr>
<td>Access:</td>
<td>RO</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>VGA_DISABLE</td>
<td>0b</td>
</tr>
<tr>
<td>Access:</td>
<td>RO</td>
<td></td>
</tr>
</tbody>
</table>

VGA Disable (VD):

0: Enable. Device 2 (IGD) claims VGA memory and IO cycles, the Sub-Class Code within Device 2 Class Code register is 00.

1: Disable. Device 2 (IGD) does not claim VGA cycles (Mem and IO), and the Sub-Class Code field within Device 2 function 0 Class Code register is 80.

BIOS Requirement: BIOS must not set this bit to 0 if the GMS field pre-allocates no memory. This bit MUST be set to 1 if Device 2 is disabled either via a fuse or fuse override (CAPID0[38] = 1) or via a register (DEVEN[3] = 0).

<table>
<thead>
<tr>
<th>REG</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>GGCLCK</td>
<td>0b</td>
</tr>
<tr>
<td>Access:</td>
<td>RO</td>
<td></td>
</tr>
</tbody>
</table>

When set to 1b, this will lock all the bits in this register, including itself.
# MGSR2GAM Message Register

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td>Mask Bits</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Mask Bits act as Write Enables for the bits[15:0] of this register.</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>MGSR2GAM Message Register 15</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For Future Use.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This bit is self clear.</td>
</tr>
<tr>
<td>14</td>
<td></td>
<td>MGSR2GAM Message Register 14</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For Future Use.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This bit is self clear.</td>
</tr>
<tr>
<td>13</td>
<td></td>
<td>MGSR2GAM Message Register 13</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For Future Use.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This bit is self clear.</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>MGSR2GAM Message Register 12</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For Future Use.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This bit is self clear.</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>MGSR2GAM Message Register 11</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For Future Use.</td>
</tr>
<tr>
<td>Register Name</td>
<td>Default Value</td>
<td>Access</td>
</tr>
<tr>
<td>------------------------------</td>
<td>---------------</td>
<td>--------</td>
</tr>
<tr>
<td>10 MGSR2GAM Message Register 10</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td>9 MGSR2GAM Message Register 9</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td>8 MGSR2GAM Message Register 8</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td>7 MGSR2GAM Message Register 7</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td>6 MGSR2GAM Message Register 6</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td>5 MGSR2GAM Message Register 5</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td>4 MGSR2GAM Message Register 4</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td>3 MGSR2GAM Message Register 3</td>
<td>0b</td>
<td>R/W</td>
</tr>
</tbody>
</table>
### MGSR2GAM_MSGREG - MGSR2GAM Message Register

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>MGSR2GAM Message Register 2</td>
<td>0b</td>
<td>R/W</td>
<td>For Future Use. This bit is self clear.</td>
</tr>
<tr>
<td>1</td>
<td>MGSR2GAM Message Register 1</td>
<td>0b</td>
<td>R/W</td>
<td>For Future Use. This bit is self clear.</td>
</tr>
<tr>
<td>0</td>
<td>MGSR2GAM Message Register 0</td>
<td>0b</td>
<td>R/W</td>
<td>Bit0 - Tail Update Ack Message. This bit is self clear.</td>
</tr>
</tbody>
</table>
### MISCCPCTL - Misc. Clocking / Reset Control Registers

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:8</td>
<td><strong>Bonus ECO bits</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td><strong>DOP clock gating enable for VEbox clks</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td><strong>DOP clock gating enable for Media clocks</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td><strong>DOP Clock gating Enable for Widi clocks</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
</tbody>
</table>

- **Register Space:** MMIO: 0/2/0
- **Project:** CHV, BSW
- **Source:** PRM
- **Default Value:** 0x00000002
- **Size (in bits):** 32
- **Address:** 09424h

Miscellaneous Clocking / Reset Control Registers.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:8</td>
<td><strong>Bonus ECO bits</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td><strong>DOP clock gating enable for VEbox clks</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td><strong>DOP clock gating enable for Media clocks</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td><strong>DOP Clock gating Enable for Widi clocks</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
</tbody>
</table>

- **DWord 0 Bit 31:8**
  - **Bonus ECO bits**
  - **Access:** R/W
  - Bonus ECO bits

- **DWord 7 Bit 7**
  - **DOP clock gating enable for VEbox clks**
  - **Access:** R/W
  - Controls the Enabling of the DOP-level Vebox (cvclk) Clock Gating via PM event messages
  - 1 - Clock gating is enabled
  - 0 - Clock gating is disabled

- **DWord 6 Bit 6**
  - **DOP clock gating enable for Media clocks**
  - **Access:** R/W
  - Controls the Enabling of the DOP-level Media (cmclk) Clock Gating via PM event messages
  - 1 - Clock gating is enabled
  - 0 - Clock gating is disabled

- **DWord 5 Bit 5**
  - **Reserved**
  - **Access:** RO
  - Made Reserved field as we dont have Media 1 in CHV, BSW
  - Controls the Enabling of the DOP-level Render (cmclk for 2nd media block) Clock Gating via PM event messages
  - 1 - Clock gating is enabled
  - 0 - Clock gating is disabled

- **DWord 4 Bit 4**
  - **Reserved**
  - **Access:**

- **DWord 3 Bit 3**
  - **DOP Clock gating Enable for Widi clocks**
  - **Access:** R/W
  - Controls the Enabling of the DOP-level Render (cwclk) Clock Gating via PM event messages
  - 1 - Clock gating is enabled
  - 0 - Clock gating is disabled
## MISCCPCTL - Misc. Clocking / Reset Control Registers

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
<th>Access</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 DOP clock gating Enable for Fix clocks (cfclk)</td>
<td>Controls the Enabling of the DOP-level Render (cfclk/cf2xclk) Clock Gating via PM event messages</td>
<td>R/W</td>
<td>1 - Clock gating is enabled&lt;br&gt;0 - Clock gating is disabled</td>
</tr>
<tr>
<td>1 L1 Clock Ungate Enabling Control During Reset</td>
<td>Control to enable/disable L1 clock gating during soft resets and FLR reset processing</td>
<td>R/W</td>
<td>1 - disable L1 clock gating during soft resets and FLR&lt;br&gt;0 - enable L1 clock gating during soft resets and FLR (default op)</td>
</tr>
<tr>
<td>0 DOP Clock Gating Enable for Render Clocks</td>
<td>Controls the Enabling of the DOP-level Render (crclk/cr2xclk) Clock Gating via PM event messages</td>
<td>R/W</td>
<td>1 - Clock gating is enabled&lt;br&gt;0 - Clock gating is disabled</td>
</tr>
</tbody>
</table>
# MISC.CTX control register

<table>
<thead>
<tr>
<th>MISCCCTXCTL - MISC. CTX control register</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 0942Ch</td>
</tr>
<tr>
<td><strong>DWORD</strong></td>
</tr>
<tr>
<td>-----------</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
## Misc. Reset Control Register

### RSTCTL - Misc. Reset Control Register

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>09420h</td>
</tr>
</tbody>
</table>

Miscellaneous reset control registers.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:4</td>
<td>Reserved</td>
<td>RO</td>
</tr>
<tr>
<td>3:2</td>
<td></td>
<td><strong>Reset Staggering Period Control</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reset assertion staggering period between reset domains during FLR and soft-resets:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>00: 24 cs clocks</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>01: 48 cs clocks</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>10: 72 cs clocks</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>11: 96 cs clocks</td>
<td></td>
</tr>
<tr>
<td>1:0</td>
<td></td>
<td><strong>Reset Residency Control</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reset assertion residency period for FLR and soft-resets.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>00: 24 cs clocks</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>01: 48 cs clocks</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>10: 96 cs clocks</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>11: 192 cs clocks</td>
<td></td>
</tr>
</tbody>
</table>
### MSG_MISC - Miscellaneous Message Register for Power Management Unit

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>16</td>
</tr>
<tr>
<td>Address:</td>
<td>08048h</td>
</tr>
</tbody>
</table>

Message registers have bit-wise masking applied for writes. The register consists of 16 bits of data in [15:0], and 16 bits of corresponding masks in [31:16].

- In order to set bit0, for example, the data would be 0x0001_0001.
- In order to clear bit0, for example, the data would be 0x0001_0000.

Note that mask bit is the data bit offset + 16. Message registers are protected from non-GT writes via the Message Channel.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>15:3</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>RO</td>
</tr>
<tr>
<td>2</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td><strong>Internal RFO Enable Ack (forwarded from MDRB to RPM to gpm)</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>From RPM on behalf of MGSR: Internal RFO Enable Acknowledgement</td>
<td></td>
</tr>
<tr>
<td></td>
<td>b0 : RFO Disable Ack (default)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>b1 : RFO Enable Ack</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td><strong>GO Acknowledgement from OAunit</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>Go Acknowledgement</td>
<td></td>
</tr>
<tr>
<td></td>
<td>b0 : Go=0 Ack &lt;default&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>b1 : Go=1 Ack</td>
<td></td>
</tr>
</tbody>
</table>

Normally the requirement is that when Go=0 request is sent, the agent has to indicate busy before sending the Go=0 acknowledgement. It can only go idle again once Go=1 is received.

For OA, however, it does not have an idle indication to PM and thus do not have to do this.
## MISR0

### MISR0 - MISR0

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:17</td>
<td><strong>Spare0</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Spare bits in case an ECO is needed.</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td><strong>Standby Threshold Periodic Enable</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Standby Threshold Periodic Enable</td>
</tr>
<tr>
<td>15:0</td>
<td></td>
<td><strong>Standby Threshold</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

### Standby Threshold

For 33MHz cz freq (200MHz, 267MHz, 320MHz, 333MHz, 400MHz, 467MHz, 533MHz)

0 = 0 us  
1 = 15.36 us  
2 = 30.72 us  
3 = 46.08 us  
FFFF ~ = 1s

For non-33MHz cz freq (350MHz, 356MHz, 360MHz, 373MHz), standby threshold will be in increments of (15.36us + (1-7)% )
### MISR1

<table>
<thead>
<tr>
<th>MISR1 - MISR1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 182044h</td>
</tr>
</tbody>
</table>

Spare1 registers. Note: Changed for CHV, BSW.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Spare1</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

Spare bits in case an ECO is needed.
MISR2

**MISR2 - MISR2**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>182048h</td>
</tr>
</tbody>
</table>

Spare2 registers. Note: Changed for CHV, BSW.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Spare2</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access:</td>
</tr>
</tbody>
</table>

Spare bits in case an ECO is needed.
## MISR3

### MISR3 - MISR3

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Spare3</strong></td>
</tr>
</tbody>
</table>

- **Default Value:** 00000000h
- **Access:** R/W

Spare bits in case an ECO is needed.

Spare3 registers. Note: Changed for CHV, BSW.
## MISR4

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>182050h</td>
</tr>
</tbody>
</table>

Spare 4 registers. Note: Changed for CHV, BSW.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Spare4</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Spare bits in case an ECO is needed.</td>
</tr>
</tbody>
</table>
**MMIO_INDEX**

<table>
<thead>
<tr>
<th>MMIO_INDEX - MMIO_INDEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space:</td>
</tr>
<tr>
<td>Project:</td>
</tr>
<tr>
<td>Source:</td>
</tr>
<tr>
<td>Default Value:</td>
</tr>
<tr>
<td>Size (in bits):</td>
</tr>
<tr>
<td>Address:</td>
</tr>
</tbody>
</table>

Contains address and target.
Punit cannot access IO space from message channel.
A 32 bit IO write to this port loads the offset of the MMIO register or offset into the GTT that needs to be accessed.
An IO Read returns the current value of this register. An 8/16 bit IO write to this register is completed but does not update this register.
This mechanism to access internal graphics MMIO registers must not be used to access VGA IO registers which are mapped through the MMIO space. VGA registers must be accessed directly through the dedicated VGA IO ports. This is used by SBIOS. It is not used by graphics driver.
This register is only accessible through the IOSF Primary bus. The base register is defined by IOBAR.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:2</td>
<td><strong>Register_offset</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This field selects GTT entry or any one of the Dword registers within the MMIO register space of this device.</td>
</tr>
<tr>
<td>1:0</td>
<td>0</td>
<td><strong>Target</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>00 = MMIO Registers, 01 = GTT, 1X = Reserved</td>
</tr>
</tbody>
</table>

---

830
Mode Register for GAB

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td><strong>Mask</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> WO</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format:</strong> Mask</td>
</tr>
<tr>
<td>15:6</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Read/Write</td>
</tr>
<tr>
<td>5:3</td>
<td></td>
<td><strong>BLB Arbitration Priority</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format:</strong> U3</td>
</tr>
<tr>
<td>2:0</td>
<td></td>
<td><strong>BCS Arbitration Priority</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format:</strong> U3</td>
</tr>
</tbody>
</table>
## Mode Register for GAC

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td>Mask</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: WO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: Mask</td>
</tr>
<tr>
<td>15:1</td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: r/w</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: r/w</td>
</tr>
</tbody>
</table>

Register Space: MMIO: 0/2/0  
Project: CHV, BSW  
Source: VideoCS  
Default Value: 0x00000000 CHV, BSW  
Access: r/w  
Size (in bits): 32  
Address: 120A0h-120A3h  
ShortName: GAC_MODE  
Valid Projects: CHV, BSW
## GAFS_MODE - Mode Register for GAFS

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>RenderCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>r/w</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Trusted Type:</td>
<td>1</td>
</tr>
<tr>
<td>Address:</td>
<td>0212Ch</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td><strong>Mask Bits</strong></td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>WO</td>
</tr>
<tr>
<td></td>
<td>Format:</td>
<td>Mask</td>
</tr>
</tbody>
</table>

Masks: These bits serve as write enables for bits 15:0. If this register is written with any of these bits clear the corresponding bit in the field 15:0 will not be modified. Reading these bits always returns 0s.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>15:11</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td>Format:</td>
<td>MBZ</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td></td>
<td>Format:</td>
<td>MBZ</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>8:2</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td>Format:</td>
<td>MBZ</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1:0</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td></td>
<td>Format:</td>
<td>MBZ</td>
</tr>
</tbody>
</table>
## MSAC - MSAC

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:21</td>
<td><strong>RESERVED</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td>20</td>
<td></td>
<td><strong>APSZ_4</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This field is used to determine the size of Aperture (GMADR) and controls affects certain bits of GMADR register.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>00000 = 128MB -&gt; GMADR.B[26:4] is hardwired to 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>00001 = 256MB -&gt; GMADR.B[27] = 0, RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>00010 = illegal (hardware will treat this as 00011)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>00011 = 512MB -&gt; GMADR.B[28:27] = 0, RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>00100-00110 = illegal (hardware will treat this as 00111)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>00111 = 1024MB -&gt; GMADR.B[29:27] = 0, RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>01000-01110 = illegal (hardware will treat this as 01111)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>01111 = 2048MB -&gt; GMADR.B[30:27] = 0, RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10000-11110 = illegal (hardware will treat this as 11111)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>11111 = 4096MB -&gt; GMADR.B[31:27] = 0, RO</td>
</tr>
<tr>
<td>19</td>
<td></td>
<td><strong>APSZ_3</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This field is used to determine the size of Aperture (GMADR) and controls affects certain bits of GMADR register.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>00000 = 128MB -&gt; GMADR.B[26:4] is hardwired to 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>00001 = 256MB -&gt; GMADR.B[27] = 0, RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>00010 = illegal (hardware will treat this as 00011)</td>
</tr>
</tbody>
</table>
### MSAC - MSAC

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>00011</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>00100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>01000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>01111</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11111</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

18 **APSZ_2**

<table>
<thead>
<tr>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0b</td>
<td>R/W</td>
</tr>
</tbody>
</table>

This field is used to determine the size of Aperture (GMADR) and controls affects certain bits of GMADR register.

- 00000 = 128MB => GMADR.B[26:4] is hardwired to 0
- 00001 = 256MB => GMADR.B[27] = 0, RO
- 00010 = illegal (hardware will treat this as 00011)
- 00011 = 512MB => GMADR.B[28:27] = 0, RO
- 00100-00110 = illegal (hardware will treat this as 00111)
- 00111 = 1024MB => GMADR.B[29:27] = 0, RO
- 01000-01110 = illegal (hardware will treat this as 01111)
- 01111 = 2048MB => GMADR.B[30:27] = 0, RO
- 10000-11110 = illegal (hardware will treat this as 11111)
- 11111 = 4096MB => GMADR.B[31:27] = 0, RO

17 **APSZ_1**

<table>
<thead>
<tr>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0b</td>
<td>R/W</td>
</tr>
</tbody>
</table>

This field is used to determine the size of Aperture (GMADR) and controls affects certain bits of GMADR register.

- 00000 = 128MB => GMADR.B[26:4] is hardwired to 0
- 00001 = 256MB => GMADR.B[27] = 0, RO
- 00010 = illegal (hardware will treat this as 00011)
- 00011 = 512MB => GMADR.B[28:27] = 0, RO
- 00100-00110 = illegal (hardware will treat this as 00111)
- 00111 = 1024MB => GMADR.B[29:27] = 0, RO
- 01000-01110 = illegal (hardware will treat this as 01111)
- 01111 = 2048MB => GMADR.B[30:27] = 0, RO
- 10000-11110 = illegal (hardware will treat this as 11111)
- 11111 = 4096MB => GMADR.B[31:27] = 0, RO

16 **APSZ_0**

<table>
<thead>
<tr>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>1b</td>
<td>R/W</td>
</tr>
</tbody>
</table>
### MSAC - MSAC

This field is used to determine the size of Aperture (GMADR) and controls affects certain bits of GMADR register.

- **00000 = 128MB** => GMADR.B[26:4] is hardwired to 0
- **00001 = 256MB** => GMADR.B[27] = 0, RO
- **00010 = illegal (hardware will treat this as 00011)**
- **00011 = 512MB** => GMADR.B[28:27] = 0, RO
- **00100-00110 = illegal (hardware will treat this as 00111)**
- **00111 = 1024MB** => GMADR.B[29:27] = 0, RO
- **01000-01110 = illegal (hardware will treat this as 01111)**
- **01111 = 2048MB** => GMADR.B[30:27] = 0, RO
- **10000-11110 = illegal (hardware will treat this as 11111)**
- **11111 = 4096MB** => GMADR.B[31:27] = 0, RO

<table>
<thead>
<tr>
<th>Field</th>
<th>Default Value</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>15:0</td>
<td>0000h</td>
<td>RO</td>
<td>Reserved</td>
</tr>
</tbody>
</table>
# MSI_CAPID_MC

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>RESERVED</strong></td>
</tr>
<tr>
<td></td>
<td>Default Value:</td>
<td>00h</td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>MODE_64B_ADDRCAP</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default Value:</td>
<td>0b</td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td>C64: 32-bit capable only</td>
<td></td>
</tr>
<tr>
<td>22:20</td>
<td>MULTIPLE_MESSAGE_ENABLE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default Value:</td>
<td>000b</td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>MME:</td>
<td>This field is RW for software compatibility, but only a single message is ever generated. System software programs this field to indicate the actual number of messages allocated to this device. This number will be equal to or less than the number actually requested. The encoding is the same as for the MMC field below.</td>
</tr>
<tr>
<td>19:17</td>
<td>MULTIPLE_MESSAGE_CAPABLE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default Value:</td>
<td>000b</td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td>MMC:</td>
<td>This device is only single message capable. System Software reads this field to determine the number of messages being requested by this device. Value: Number of requests 000: 1 001-111: Reserved</td>
</tr>
</tbody>
</table>
### MSI_CAPID_MC - MSI_CAPID_MC

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>16</strong></td>
<td><strong>MSI_ENABLE</strong></td>
</tr>
<tr>
<td>Default Value</td>
<td>0b</td>
</tr>
<tr>
<td>Access</td>
<td>R/W</td>
</tr>
<tr>
<td><strong>MSIE:</strong></td>
<td>If set, MSI is enabled and traditional interrupts are not used to generate interrupts. PCICMDSTS.BME must be set for an MSI to be generated.</td>
</tr>
<tr>
<td>0 : MSI interrupts are disabled.</td>
<td>1 : MSI interrupts are enabled. Permits sending an MSI interrupt.</td>
</tr>
<tr>
<td><strong>15:8</strong></td>
<td><strong>POINTER_TO_NEXT_CAPABILITY</strong></td>
</tr>
<tr>
<td>Default Value</td>
<td>B0h</td>
</tr>
<tr>
<td>Access</td>
<td>R/W Once</td>
</tr>
<tr>
<td>Points to the next item in the list(b0=VCID support).</td>
<td>This register should be programmed by BIOS during boot-up. Once written, this register becomes Read Only. This register can only be cleared by a Reset.</td>
</tr>
<tr>
<td><strong>7:0</strong></td>
<td><strong>CAPABILITY_ID</strong></td>
</tr>
<tr>
<td>Default Value</td>
<td>05h</td>
</tr>
<tr>
<td>Access</td>
<td>RO</td>
</tr>
<tr>
<td>CAPID: Indicates an MSI capability</td>
<td></td>
</tr>
</tbody>
</table>
### MTRR Capability Register 0

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:11</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0000000000000000000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td><strong>Write Combining Support</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 1b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td>0:</td>
<td>Write Combining (WC) memory type is not supported.</td>
</tr>
<tr>
<td></td>
<td>1:</td>
<td>Write Combining (WC) memory type is supported.</td>
</tr>
<tr>
<td></td>
<td>GFX Implementation: More details on memory type section however WC support in GFX looks like streamlining non-cacheable accesses. This is the existing UC concept used in GFX architecture.</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td><strong>Fixed Range MTRRs Support</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 1b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td>0:</td>
<td>No Fixed range MTRRs are supported.</td>
</tr>
<tr>
<td></td>
<td>1:</td>
<td>Fixed Range MTRRs (IA32_MTRR_FIX64K_00000 through IA32_MTRR_FIX4K_0F8000) are supported.</td>
</tr>
<tr>
<td>7:0</td>
<td></td>
<td><strong>Variable Range MTRR Count</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0Ah</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td>Indicates the number of variable ranges implemented.</td>
<td></td>
</tr>
</tbody>
</table>
# MTRR Capability Register 1

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong></td>
<td>MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong></td>
<td>CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong></td>
<td>PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong></td>
<td>0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong></td>
<td>32</td>
</tr>
<tr>
<td><strong>Address:</strong></td>
<td>0F104h</td>
</tr>
</tbody>
</table>

Register to define MTRR - range register capabilities

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>MTRR Capability Register 1 Reserved</strong></td>
</tr>
</tbody>
</table>

- **Default Value:** 00000000h
- **Access:** RO
- **Bit[63:32]: Reserved.**
### MTRR Default Type Register 0

**MTRR_DT_0 - MTRR Default Type Register 0**

- **Register Space:** MMIO: 0/2/0
- **Project:** CHV, BSW
- **Source:** PRM
- **Default Value:** 0x00000000
- **Size (in bits):** 32
- **Address:** 0F108h

Register to define MTRR range register capabilities.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:12</td>
<td><strong>Reserved</strong></td>
<td>00000000000000000000b</td>
<td>RO</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td><strong>Reserved</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td><strong>Fixed Range MTRR Enable/Disable</strong></td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>9:8</td>
<td><strong>Reserved</strong></td>
<td>00b</td>
<td></td>
</tr>
<tr>
<td>7:0</td>
<td></td>
<td><strong>Default Memory Type</strong></td>
<td>00h</td>
<td>R/W</td>
</tr>
</tbody>
</table>

- **Default Value:**
  - 00000000000000000000b
  - 00h

- **Access:**
  - RO
  - R/W

### Description

- **Fixed Range MTRR Enable/Disable**
  - **Default Value:** 0b
  - **Access:** R/W
  - 0: Disable fixed-range MTRRs.
  - 1: Enable fixed-range MTRRs.
  - When the fixed-range MTRRs are enabled, they take priority over the variable-range MTRRs when overlaps in ranges occur. If the fixed-range MTRRs are disabled, the variable range MTRRs can still be used and can map the range ordinarily covered by the fixed-range MTRRs.
  - **GFX Implementation:** GFX uses this field as a specific enable/disable for fixed range MTRRs.

- **Default Memory Type**
  - **Default Value:** 00h
  - **Access:** R/W
  - Indicates default memory type used for physical memory address ranges that do not have a memory type specified for them by an MTRR. Legal values for this field are 0, 1, 4, 5, and 6.
  - **GFX Implementation:** GFX uses this field to assign memory regions that are not assigned as part of the fixed and variable range registers.
MTRR Default Type Register 1

<table>
<thead>
<tr>
<th>MTRR_DT_1 - MTRR Default Type Register 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 0F10Ch</td>
</tr>
</tbody>
</table>

Register to define MTRR - range register capabilities.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>MTRR Default Type Register 1 Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[63:32]: Reserved.</td>
</tr>
</tbody>
</table>
## MT Virtual Page Address Registers

### MTTLB_VA - MT Virtual Page Address Registers

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>RenderCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>RO</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Trusted Type:</td>
<td>1</td>
</tr>
</tbody>
</table>

**Address:** 04800h-04803h

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:12</td>
<td><strong>Address</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: GraphicsAddress[31:12]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Page virtual address.</td>
</tr>
<tr>
<td>0</td>
<td>11:0</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
</tbody>
</table>
NOP Identification Register

<table>
<thead>
<tr>
<th>NOPID - NOP Identification Register</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: RenderCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Trusted Type: 1</td>
</tr>
<tr>
<td>Address: 02094h</td>
</tr>
<tr>
<td>Name: NOP Identification Register</td>
</tr>
<tr>
<td>ShortName: NOPID_VCSUNIT0</td>
</tr>
<tr>
<td>Address: 12094h-12097h</td>
</tr>
<tr>
<td>Name: NOP Identification Register</td>
</tr>
<tr>
<td>ShortName: NOPID_VCSUNIT</td>
</tr>
<tr>
<td>Address: 1A094h-1A097h</td>
</tr>
<tr>
<td>Name: NOP Identification Register</td>
</tr>
<tr>
<td>ShortName: NOPID_VCSUNIT1</td>
</tr>
<tr>
<td>Address: 22094h-22097h</td>
</tr>
<tr>
<td>Name: NOP Identification Register</td>
</tr>
<tr>
<td>ShortName: NOPID_BCSUNIT</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Description</th>
<th>Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access: RW</td>
<td>CHV, BSW</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:22</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td>Format:</td>
<td>MBZ</td>
</tr>
</tbody>
</table>

|       | 21:0   | Reserved    |
## P2GCONTROL

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>1300F0h</td>
</tr>
</tbody>
</table>

Punit to Gunit Control. Punit will context save/restore this register.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td><strong>Reserved</strong></td>
<td><strong>RO</strong></td>
</tr>
<tr>
<td>0</td>
<td>31:1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td><strong>P2GCONTROL_POLICY</strong></td>
<td><strong>R/W</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0: Let Gunit do the context save. (Power-On default)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1: Context save will be handled by Punit and driver.</td>
<td></td>
</tr>
</tbody>
</table>
## PAGE_FAULT_MODE

<table>
<thead>
<tr>
<th>Register Space: MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 0E454h</td>
</tr>
<tr>
<td>Name: PAGE_FAULT_MODE</td>
</tr>
<tr>
<td>ShortName: PAGE_FAULT_MODE</td>
</tr>
<tr>
<td>Valid Projects: CHV, BSW</td>
</tr>
</tbody>
</table>

This is a basic register template

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:8</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 000000000000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td>7:6</td>
<td><strong>FAULT_MODE</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

**Fault Model:** Applicable only in advanced context
- "00": Fault & Hang. Same mode as gen7.5
- "01": Fault & Halt
- "10": Fault & Continue & Switch
- "11": Reserved

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>00b</td>
<td>[Default]</td>
</tr>
<tr>
<td>01b</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>5:0</td>
<td><strong>Reserved</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Project: CHV, BSW</td>
<td></td>
</tr>
</tbody>
</table>
Page Directory Pointer Descriptor - PDP0/PML4/PASID

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000, 0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>64</td>
</tr>
<tr>
<td>Address:</td>
<td>02270h-02277h</td>
</tr>
<tr>
<td>Name:</td>
<td>Page Directory Pointer Descriptor - PDP0/PML4/PASID</td>
</tr>
<tr>
<td>ShortName:</td>
<td>PDP0_RCSUNIT</td>
</tr>
</tbody>
</table>

| Address:       | 12270h-12277h |
| Name:          | Page Directory Pointer Descriptor - PDP0/PML4/PASID |
| ShortName:     | PDP0_VCSUNIT0 |

| Address:       | 1A270h-1A277h |
| Name:          | Page Directory Pointer Descriptor - PDP0/PML4/PASID |
| ShortName:     | PDP0_VECSUNIT |

| Address:       | 1C270h-1C277h |
| Name:          | Page Directory Pointer Descriptor - PDP0/PML4/PASID |
| ShortName:     | PDP0_VCSUNIT1 |

| Address:       | 22270h-22277h |
| Name:          | Page Directory Pointer Descriptor - PDP0/PML4/PASID |
| ShortName:     | PDP0_BCSUNIT |

**PDP0/PML4/PASID**: This register can contain three values which depend on the element descriptor definition. 

**PASID[19:0]**: Populated in the first 20 bits of the register and selected when Advanced Context flag is set in the element descriptor in exelist mode of submission. This is not valid in ring buffer mode of scheduling.

**PML4[38:12]**: Pointer to base address of PML4 and selected when Legacy Context flag is set and 64b address support is selected. **PDP0[38:12]**: Pointer to one of the four page directory pointer (lowest) and defines the first 0-1GB of memory mapping. This is valid when Legacy Context Flag is set and 64bit virtual addressing is not supported. Note: This is a guest physical address.

**Programming Notes**

_**Execlist Based Scheduling**_: SW should update PDP0/12/3 registers in context image with proper values before submitting the context to HW in execlist mode of scheduling. HW restores these registers as part of context restore to set the PPGTT access accordingly. PPGTT is always enabled in advanced context mode of execlist based scheduling and can be disabled only in legacy context mode. Privilege Access Bit in Element Descriptor controls the PPGTT enabling in legacy context mode.
**PDP0 - Page Directory Pointer Descriptor - PDP0/PML4/PASID**

*Ring Buffer Based Scheduling:* A write via MMIO to PDP0_DESCRIPTOR (lower Dword) triggers the Page Directory Restore in HW when PPGTT is enabled. SW should ensure PDP1/2/3 registers are programmed appropriately prior to programming PDP0. PDP0_DESCRIPTOR lower dword should be programmed at the end. Per-Process GTT Enable Bit in GFX_MODE register controls the PPGTT enabling and disabling. Programming Per-Process GTT Enable Bit in GFX_MODE register doesn’t enable/disable the PPGTT translation of memory access immediately; the change comes in to affect only when the Page Directory registers are programmed. Programming Per-Process GTT Enable Bit in GFX_MODE register bit must be followed by programming Page Directory Registers in order to enable or disable the PPGTT translation of memory access. PDP* DESCRIPTOR registers must always be programmed through MI_LOAD_REGISTER_IMMEDIATE command in ring buffer with PDP0_DESCRIPTOR lower dword written at the end. PDP0/12/3 registers are context save restored. PDP descriptors are power context save restored in VCS, BCS and VECS engines. PDP descriptors are context save restored per render context in RCS and must be programmed following MI_SET_CONTEXT command, in case of PDP descriptors programmed without context set (MI_SET_CONTEXT) will get lost on C6 entry/exit. PDP descriptor registers should be programmed after ensuring the pipe is completely flushed and TLB’s invalidated.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Project, Access, Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>63</td>
<td>PD Load Busy</td>
<td>CHV, BSW, RO, Valid</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This read-only field gets set when PDP0 is written to indicating Page Directory Restore activity is in progress and will get reset once the activity is completed.</td>
<td></td>
</tr>
<tr>
<td>62:0</td>
<td></td>
<td>PDP0 Descriptor</td>
<td>CHV, BSW, RO</td>
</tr>
</tbody>
</table>

**CHV, BSW**
### Page Directory Pointer Descriptor - PDP1

<table>
<thead>
<tr>
<th>PDP1 - Page Directory Pointer Descriptor - PDP1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000, 0x00000000</td>
</tr>
<tr>
<td>Access: R/W</td>
</tr>
<tr>
<td>Size (in bits): 64</td>
</tr>
</tbody>
</table>

| Address: 02278h-0227Fh |
| Name: Page Directory Pointer Descriptor - PDP1 |
| ShortName: PDP1_RCSUNIT |

| Address: 12278h-1227Fh |
| Name: Page Directory Pointer Descriptor - PDP1 |
| ShortName: PDP1_VCSUNIT0 |

| Address: 1A278h-1A27Fh |
| Name: Page Directory Pointer Descriptor - PDP1 |
| ShortName: PDP1_VECSUNIT |

| Address: 1C278h-1C27Fh |
| Name: Page Directory Pointer Descriptor - PDP1 |
| ShortName: PDP1_VCSUNIT1 |

| Address: 22278h-2227Fh |
| Name: Page Directory Pointer Descriptor - PDP1 |
| ShortName: PDP1_BCSUNIT |

**PDP1[38:12]:** Pointer to one of the four page directory pointer (lowest+1) and defines the first 1-2GB of memory mapping. This is valid when Legacy Context Flag is set and 64bit virtual addressing is not supported. 

*Note: This is a guest physical address.*

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>63:0</td>
<td><strong>PDP1 Descriptor</strong></td>
</tr>
</tbody>
</table>
## Page Directory Pointer Descriptor - PDP2

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000, 0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>64</td>
</tr>
<tr>
<td>Address:</td>
<td>02280h-02287h</td>
</tr>
<tr>
<td>Name:</td>
<td>Page Directory Pointer Descriptor - PDP2</td>
</tr>
<tr>
<td>ShortName:</td>
<td>PDP2_RCSUNIT</td>
</tr>
</tbody>
</table>

| Address:       | 12280h-12287h |
| Name:          | Page Directory Pointer Descriptor - PDP2 |
| ShortName:     | PDP2_VCSUNIT0 |

| Address:       | 1A280h-1A287h |
| Name:          | Page Directory Pointer Descriptor - PDP2 |
| ShortName:     | PDP2_VECSUNIT |

| Address:       | 1C280h-1C287h |
| Name:          | Page Directory Pointer Descriptor - PDP2 |
| ShortName:     | PDP2_VCSUNIT1 |

| Address:       | 22280h-22287h |
| Name:          | Page Directory Pointer Descriptor - PDP2 |
| ShortName:     | PDP2_BCSUNIT |

**PDP2[38:12]:** Pointer to one of the four page directory pointer (lowest+2) and defines the first 2-3GB of memory mapping. This is valid when Legacy Context Flag is set and 64bit virtual addressing is not supported. Note: This is a guest physical address.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>63:0</td>
<td><strong>PDP2 Descriptor</strong></td>
</tr>
</tbody>
</table>
## Page Directory Pointer Descriptor - PDP3

<table>
<thead>
<tr>
<th>PDP3 - Page Directory Pointer Descriptor - PDP3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000, 0x00000000</td>
</tr>
<tr>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 64</td>
</tr>
<tr>
<td><strong>Address:</strong> 02288h-0228Fh</td>
</tr>
<tr>
<td><strong>Name:</strong> Page Directory Pointer Descriptor - PDP3</td>
</tr>
<tr>
<td><strong>ShortName:</strong> PDP3_RCSUNIT</td>
</tr>
<tr>
<td><strong>Address:</strong> 12288h-1228Fh</td>
</tr>
<tr>
<td><strong>Name:</strong> Page Directory Pointer Descriptor - PDP3</td>
</tr>
<tr>
<td><strong>ShortName:</strong> PDP3_VCSUNIT0</td>
</tr>
<tr>
<td><strong>Address:</strong> 1A288h-1A28Fh</td>
</tr>
<tr>
<td><strong>Name:</strong> Page Directory Pointer Descriptor - PDP3</td>
</tr>
<tr>
<td><strong>ShortName:</strong> PDP3_VECSUNIT</td>
</tr>
<tr>
<td><strong>Address:</strong> 1C288h-1C28Fh</td>
</tr>
<tr>
<td><strong>Name:</strong> Page Directory Pointer Descriptor - PDP3</td>
</tr>
<tr>
<td><strong>ShortName:</strong> PDP3_VCSUNIT1</td>
</tr>
<tr>
<td><strong>Address:</strong> 22288h-2228Fh</td>
</tr>
<tr>
<td><strong>Name:</strong> Page Directory Pointer Descriptor - PDP3</td>
</tr>
<tr>
<td><strong>ShortName:</strong> PDP3_BCSUNIT</td>
</tr>
</tbody>
</table>

**PDP3[38:12]:** Pointer to one of the four page directory pointer (lowest+3) and defines the first 3-4GB of memory mapping. This is valid when Legacy Context Flag is set and 64bit virtual addressing is not supported. *Note: This is a guest physical address.*

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>63:0</td>
<td><strong>PDP3 Descriptor</strong></td>
</tr>
</tbody>
</table>
# Page Request Queue Address Register 0

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:12</td>
<td>Page Request Queue Base Register</td>
<td>00000h</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This field points to the base of 4KB aligned invalidation request queue. Hardware may ignore and not implement bits 63:HAW, where HAW is the host address width. Reads of this field return the value that was last programmed to it.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11:3</td>
<td></td>
<td>Reserved</td>
<td>000000000b</td>
<td>RO</td>
</tr>
<tr>
<td>2:0</td>
<td></td>
<td>Queue Size</td>
<td>000b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This field specifies the size of the page request queue. A value of X in this field indicates a page request queue of (2X) 4KB pages. The number of entries in the invalidation queue is 2(X+8).</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Register to configure the base address and size of the page request queue.

Register Space: MMIO: 0/2/0
Source: PRM
Default Value: 0x00000000
Size (in bits): 32
Address: 0F0D0h
## Page Request Queue Address Register 1

### PAGEREQ_QADDR_1 - Page Request Queue Address Register 1

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits)</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0F0D4h</td>
</tr>
</tbody>
</table>

Register to configure the base address and size of the page request queue.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Page Request Queue Base Register</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This field points to the base of 4KB aligned invalidation request queue. Hardware may ignore and not implement bits 63:HAW, where HAW is the host address width. Reads of this field return the value that was last programmed to it.</td>
</tr>
</tbody>
</table>
## Page Request Queue Head Register 0

<table>
<thead>
<tr>
<th>PAGEREQ_QHEAD_0 - Page Request Queue Head Register 0</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 0F0C0h</td>
</tr>
</tbody>
</table>

Register indicating the page request queue head.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:19</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 000000000000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>18:4</td>
<td>Queue Head</td>
<td>Default Value: 000000000000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Specifies the offset (128-bit aligned) to the page request queue for the command that is processed next by software. GFX implementation: GFX has to read the content of the Head pointer as tail pointer gets close to it to prevent overflows in page request queue.</td>
</tr>
<tr>
<td>3:0</td>
<td><strong>Reserved</strong></td>
<td>Default Value: 0h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
</tbody>
</table>
# Page Request Queue Head Register 1

<table>
<thead>
<tr>
<th><strong>PAGEREQ_QHEAD_1 - Page Request Queue Head Register 1</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 0F0C4h</td>
</tr>
</tbody>
</table>

Register indicating the page request queue head.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Page Request Queue Head Register 1 Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[63:32]: Reserved.</td>
</tr>
</tbody>
</table>
Page Request Queue Tail Register 0

**Register Space:** MMIO: 0/2/0  
**Project:** CHV, BSW  
**Source:** PRM  
**Default Value:** 0x00000000  
**Size (in bits):** 32  
**Address:** 0F0C8h

Register indicating the page request queue tail.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:1</td>
<td><strong>Queue Tail</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000000000000000000000000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>
|       |     | Bit[31:19]: Reserved.  
|       |     | Bit[18:4]:  
|       |     | Specifies the offset (128-bit aligned) to the page request queue for the request that is written next by hardware.  
|       |     | GFX Implementation: GT manages the tail pointer value as part of page requests. The value can be acquired as part of the RC6 exit.  
|       |     | Bit[3:1]: Reserved.  |

| 0 | **Valid Bit** |
|   | Default Value: 0b  |
|   | Access: R/W  |

This bit can only be cleared by SW, which also clears the other fields.
# Page Request Queue Tail Register 1

**PAGEREQ_QTAIL_1 - Page Request Queue Tail Register 1**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0F0CCh</td>
</tr>
</tbody>
</table>

Register indicating the page request queue tail.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Page Request Queue Tail Register 1 Reserved</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Default Value:</th>
<th>00000000h</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access:</td>
<td>RO</td>
</tr>
</tbody>
</table>

Bit[63:32]: Reserved.
# PAK_Stream-Out Report (Errors)

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:22</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
<tr>
<td>21</td>
<td></td>
<td>Incorrect IntraMBFlag in I-slice(AVCf)</td>
</tr>
<tr>
<td>20</td>
<td></td>
<td>Out of Range Symbol Code(AVC/mpeg2)</td>
</tr>
<tr>
<td>19</td>
<td></td>
<td>Incorrect MBType(AVC/mpeg2)</td>
</tr>
<tr>
<td>18</td>
<td></td>
<td>Motion Vectors are not inside the frame boundary(mpeg2)</td>
</tr>
<tr>
<td>17</td>
<td></td>
<td>Scale code is zero(mpeg2)</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td>Incorrect DCTtype for given motionType(mpeg2)</td>
</tr>
<tr>
<td>15:8</td>
<td></td>
<td>MB Y-position</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This field indicates Macro Block(MB) Y- position where an error occurred</td>
</tr>
<tr>
<td></td>
<td></td>
<td>while encoding.</td>
</tr>
<tr>
<td>7:0</td>
<td></td>
<td>MB X-position</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This field indicates Macro Block(MB) X- position where an error occurred</td>
</tr>
</tbody>
</table>
### PAK_WARN - PAK_Stream-Out Report (Warnings)

<table>
<thead>
<tr>
<th>Register Space</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>VideoCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>RO</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Trusted Type:</td>
<td>1</td>
</tr>
<tr>
<td>Address:</td>
<td>128E4h</td>
</tr>
<tr>
<td>Valid Projects:</td>
<td>CHV, BSW</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:22</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Project:</strong> All</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format:</strong> MBZ</td>
</tr>
<tr>
<td>21</td>
<td></td>
<td><strong>Skip Run &gt; 8192 (AVC)</strong></td>
</tr>
<tr>
<td>20</td>
<td></td>
<td><strong>Incorrect SkipMB (AVC and mpeg2)</strong></td>
</tr>
<tr>
<td>19</td>
<td></td>
<td><strong>Incorrect MV difference for dual-prime MB (mpeg2)</strong></td>
</tr>
<tr>
<td>18</td>
<td></td>
<td><strong>End of Slice signal missing on last MB of a Row (mpeg2)</strong></td>
</tr>
<tr>
<td>17</td>
<td></td>
<td><strong>Incorrect DCT type for field picture</strong></td>
</tr>
<tr>
<td>16</td>
<td></td>
<td><strong>MVs are not within defined range by fcode</strong></td>
</tr>
<tr>
<td>15:8</td>
<td></td>
<td><strong>MB Y-position</strong></td>
</tr>
<tr>
<td>7:0</td>
<td></td>
<td><strong>MB X-position</strong></td>
</tr>
</tbody>
</table>
PAK Report Running Status

### PAK_REPORT_STAT - PAK Report Running Status

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>VideoCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>RO</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Trusted Type:</td>
<td>1</td>
</tr>
<tr>
<td>Address:</td>
<td>128ECh</td>
</tr>
<tr>
<td>Valid Projects:</td>
<td>CHV, BSW</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:1</td>
<td>Reserved</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td><strong>PAK Status</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>PAK engine is IDLE</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>PAK engine is currently generating bit stream.</td>
<td></td>
</tr>
</tbody>
</table>
## PAT Index

### PAT_INDEX - PAT Index

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:10</td>
<td><strong>Reserved</strong>&lt;br&gt;Default Value: 0000000000000000000000b</td>
</tr>
<tr>
<td>9:8</td>
<td>Class of Service&lt;br&gt;Default Value: 00b&lt;br&gt;This field controls the Class of Service sent to the LLC to determine which sub-set of Ways the surface will be stored in. The allocation of certain LLC ways to different class of service settings is a project dependent decision and listed in the PRM:&lt;br&gt;00: Class0&lt;br&gt;01: Class1&lt;br&gt;10: Class2&lt;br&gt;11: Class3</td>
<td></td>
</tr>
<tr>
<td>7:6</td>
<td><strong>Reserved</strong>&lt;br&gt;Default Value: 00b</td>
<td></td>
</tr>
</tbody>
</table>
| 5:4   | LRU AGE<br>Default Value: 00b<br>00: Take the age value from Uncore CRs<br>01: Assign the age of "0"
10: Do not change the age on a hit<br>11: Assign the age of "3" |
| 3:2   | Target Cache<br>Default Value: 00b<br>00: eLLC only<br>01: LLC only<br>10: LLC/eLLC allowed<br>11: LLC/eLLC allowed |
| 1:0   | Mem Type<br>Default Value: 11b<br>00: Uncacheable(UC)<br>01: Write Combining(WC)<br>10: Write through(WT)<br>11: Write back(WB) |
## PAT_INDEX_H - PAT Index High

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>PAT Index High</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 03030303h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

**Description**

- Bit[31:24]: PAT Index#7: Index#7 definition for page tables. (See bit[7:0] for definition.)
- Bit[23:16]: PAT Index#6: Index#6 definition for page tables. (See bit[7:0] for definition.)
- Bit[15:8]: PAT Index#5: Index#5 definition for page tables. (See bit[7:0] for definition.)
- Bit[7:0]: PAT Index#4: Index#4 definition for page tables.
- Bit[7]: Reserved.
- Bit[6]: Snoop Required [CHV, BSW Only]
  - 1: System agent will snoop the IA cores
  - 0: System agent will not snoop the IA cores
- Bit[5:4]: (See below.)
  - 00b: Age is 0.
  - 01b: Age is 1.
  - 10b: Age is 2.
  - 11b: Age is 3.
- Bit[3:2]: (See below.)
  - 00b: eLLC only.
  - 01b: LLC only.
  - 10b: LLC and eLLC allowed.
  - 11b: L3, LLC, and eLLC are allowed.
- Bit[1:0]: (See below):
  - 00b: Uncacheable (UC).
  - 01b: Write Combining (WC).
  - 10b: Write Through (WT).
  - 11b: Write Back (WB).
## PAT Index Low

### PAT_INDEX_L - PAT Index Low

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x03030303</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>040E0h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>PAT Index Low</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 03030303h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

**Description**

- Bit[31:24]: PAT Index#3: Index#3 definition for page tables. (See bit[7:0] for definition.)
- Bit[23:16]: PAT Index#2: Index#2 definition for page tables. (See bit[7:0] for definition.)
- Bit[15:8]: PAT Index#1: Index#1 definition for page tables. (See bit[7:0] for definition.)
- Bit[7:0]: PAT Index#0: Index#0 definition for page tables.
- Bit[7]: Reserved.
- Bit[6]: Snoop Required [CHV, BSW Only]
  - 1: System agent will snoop the IA cores
  - 0: System agent will not snoop the IA cores
- Bit[5:4]: (See below.)
  - 00b: Age is 0.
  - 01b: Age is 1.
  - 10b: Age is 2.
  - 11b: Age is 3.
- Bit[3:2]: (See below.)
  - 00b: eLLC only.
  - 01b: LLC only.
  - 10b: LLC and eLLC allowed.
  - 11b: L3, LLC, and eLLC are allowed.
- Bit[1:0]: (see below):
  - 00b: Uncacheable (UC).
  - 01b: Write Combining (WC).
  - 10b: Write Through (WT).
  - 11b: Write Back (WB).
## PCBR

### PCBR - PCBR

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
</tbody>
</table>

**Address:** 182120h

This provides a base address for context save/restore of GT and Media power context to DRAM. Gunit adds the GAM provided offsets to this base register for power context reads and writes. GTLC stores Power Context in DRAM. The BIOS is expected to program this register and ensure proper allocation within Gfx stolen memory.

Removing bits 63:32 since this will always be below 4GB.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:12</td>
<td><strong>Power_Context_Address</strong></td>
<td>0000000h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4KB aligned address. Locked with bit 0.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11:1</td>
<td></td>
<td><strong>Reserved</strong></td>
<td>000h</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td><strong>Power_Context_Register_Lock</strong></td>
<td>0b</td>
<td>R/W Lock</td>
</tr>
</tbody>
</table>

Writing a ‘1’ to this register locks this bit - preventing further updates. The Power Context Address bits are also locked.
# PCICMD_STS

## Register Space
- **PCI:** 0/2/0

## Project
- CHV, BSW

## Source
- PRM

## Default Value
- 0x00100000

## Size (in bits)
- 32

## Address
- 00004h

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:21</td>
<td><strong>RESERVED</strong></td>
<td>000h</td>
<td>RO</td>
</tr>
<tr>
<td>20</td>
<td></td>
<td><strong>CAPABILITY_LIST</strong></td>
<td>1b</td>
<td>RO</td>
</tr>
<tr>
<td>19</td>
<td></td>
<td><strong>INTERRUPT_STATUS</strong></td>
<td>0b</td>
<td></td>
</tr>
<tr>
<td>18:16</td>
<td></td>
<td><strong>RESERVED</strong></td>
<td>000b</td>
<td>RO</td>
</tr>
<tr>
<td>15:11</td>
<td></td>
<td><strong>RESERVED</strong></td>
<td>00h</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td><strong>INTERRUPT_DISABLE</strong></td>
<td>0b</td>
<td>R/W</td>
</tr>
</tbody>
</table>

- **ID:** 0 : Legacy interrupt message is enabled.
- **ID:** 1 : Disables legacy interrupt message generation on IOSF Sideband. Note : The interrupt status is not blocked from being reflected in PCISTS.IS.
- Note: MSI interrupt generation : (PCISTS.IS and PCICMD.BME and MSI_CAPID.MSIE) changes from 0 to 1.
- Note: Message bus interrupt assert is sent : (PCISTS.IS and ~PCICMD.ID and ~MSI_CAPID.MSIE) changes from 0 to 1.
- Note: Message bus interrupt de-assert is sent : (PCISTS.IS and ~PCICMD.ID and ~MSI_CAPID.MSIE) changes from 1 to 0.
### PCICMD_STS - PCICMD_STS

<table>
<thead>
<tr>
<th>Bit Positions</th>
<th>Field</th>
<th>Default Value</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>9:3</td>
<td>RESERVED</td>
<td>00h</td>
<td>RO</td>
<td>Default Value: 00h. Access: RO. Reserved</td>
</tr>
<tr>
<td>2</td>
<td>BUS_MASTER_ENABLE</td>
<td>0b</td>
<td>R/W</td>
<td>Default Value: 0b. Access: R/W. BME: (BME and MAE are observed. But context save/restore can occur.) 0 : Blocks the sending of MSI interrupts. 1 : Permits the sending of MSI interrupts</td>
</tr>
<tr>
<td>1</td>
<td>Reserved</td>
<td></td>
<td></td>
<td>Access: R/W. Reserved</td>
</tr>
<tr>
<td>0</td>
<td>IO_SPACE_ENABLE</td>
<td>0b</td>
<td>R/W</td>
<td>Default Value: 0b. Access: R/W. IOSE: 0 : I/O space is disabled. IORD and IOWr cycles will not be claimed. 1 : I/O space is enabled. VGA_IO and Gfx_IOBAR are checked. If an IORD/IOWR matches (VGA IO address range or GFX_IOBAR), the cycle will be claimed. Care should be taken in setting up GFX_IOBAR that more than 1 match is not made as this will result in unpredictable behavior. VGA_IO : Address[15:0] is checked to determine if it falls in the VGA IO range. (The VGA IO range is 03B0h - 03BBh and 03C0h - 03DFh.) Gfx_IOBAR : Address[15:3] is compared to GFX_IOBAR[15:3].</td>
</tr>
</tbody>
</table>
## PCU INTERRUPT ENABLE REGISTER

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:26</td>
<td>UNUSED0</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>PCU_MAILBOX_EVT</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>PCU Pcode 2 driver mailbox event</td>
<td></td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>PCU_THERMAL_EVT</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>PCU thermal event</td>
<td></td>
</tr>
<tr>
<td></td>
<td>23:0</td>
<td>UNUSED1</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access:</td>
<td></td>
</tr>
</tbody>
</table>

This table indicates which events are mapped to each bit of the GT interrupt 1 register. The IER enabled PCU interrupt IIR sticky bits are ORed together to generate PCU interrupt pending bit in the master interrupt control register.
## PCU INTERRUPT IDENTITY REGISTER

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:26</td>
<td>UNUSED0</td>
<td>R/W One Clear</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td></td>
<td>PCU_MAILBOX_EVT</td>
<td>R/W One Clear</td>
<td>PCU Pcode 2 driver mailbox event</td>
</tr>
<tr>
<td>24</td>
<td></td>
<td>PCU_THERMAL_EVT</td>
<td>R/W One Clear</td>
<td>PCU thermal event</td>
</tr>
<tr>
<td>23:0</td>
<td></td>
<td>UNUSED1</td>
<td>R/W One Clear</td>
<td></td>
</tr>
</tbody>
</table>
# PCU INTERRUPT MASK REGISTER

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:26</td>
<td>UNUSED0</td>
<td>R/W</td>
</tr>
<tr>
<td>25</td>
<td></td>
<td>PCU_MAILBOX_EVT</td>
<td>1b R/W</td>
</tr>
<tr>
<td></td>
<td>Default Value:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PCU Pcode 2 driver mailbox event</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>23:0</td>
<td>PCU_THERMAL_EVT</td>
<td>1b R/W</td>
</tr>
<tr>
<td></td>
<td>Default Value:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PCU thermal event</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>UNUSED1</td>
<td>Access:</td>
<td>R/W</td>
</tr>
</tbody>
</table>
PCU INTERRUPT STATUS REGISTER

<table>
<thead>
<tr>
<th>PCU_INTERRUPT_ISR - PCU INTERRUPT STATUS REGISTER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 444E0h-444E3h</td>
</tr>
</tbody>
</table>

This table indicates which events are mapped to each bit of the GT interrupt 1 register. The IER enabled PCU interrupt IIR sticky bits are ORed together to generate PCU interrupt pending bit in the master interrupt control register.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
<th>Mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:26</td>
<td>UNUSED0</td>
<td></td>
<td>RO</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td></td>
<td>PCU_MAILBOX_EVT</td>
<td></td>
<td>RO</td>
<td>PCU Pcode 2 driver mailbox event</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td></td>
<td>PCU_THERMAL_EVT</td>
<td></td>
<td>RO</td>
<td>PCU thermal event</td>
</tr>
<tr>
<td>23:0</td>
<td></td>
<td>UNUSED1</td>
<td></td>
<td>RO</td>
<td></td>
</tr>
</tbody>
</table>
### Pending Head Pointer Register

<table>
<thead>
<tr>
<th><strong>UHPRTR - Pending Head Pointer Register</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 02134h-02137h</td>
</tr>
<tr>
<td><strong>Name:</strong> Pending Head Pointer Register</td>
</tr>
<tr>
<td><strong>ShortName:</strong> UHPRTR_RCSUNIT</td>
</tr>
<tr>
<td><strong>Address:</strong> 12134h-12137h</td>
</tr>
<tr>
<td><strong>Name:</strong> Pending Head Pointer Register</td>
</tr>
<tr>
<td><strong>ShortName:</strong> UHPRTR_VCSUNIT0</td>
</tr>
<tr>
<td><strong>Address:</strong> 1A134h-1A137h</td>
</tr>
<tr>
<td><strong>Name:</strong> Pending Head Pointer Register</td>
</tr>
<tr>
<td><strong>ShortName:</strong> UHPRTR_VCSUNIT1</td>
</tr>
<tr>
<td><strong>Address:</strong> 1C134h-1C137h</td>
</tr>
<tr>
<td><strong>Name:</strong> Pending Head Pointer Register</td>
</tr>
<tr>
<td><strong>ShortName:</strong> UHPRTR_VCSUNIT</td>
</tr>
<tr>
<td><strong>Address:</strong> 22134h-22137h</td>
</tr>
<tr>
<td><strong>Name:</strong> Pending Head Pointer Register</td>
</tr>
<tr>
<td><strong>ShortName:</strong> UHPRTR_BCSUNIT</td>
</tr>
</tbody>
</table>

### Programming Notes

Once SW uses UHPRTR to preempt the existing workload, should explicitly program MI_SET_CONTEXT to save the preempted context status before submitting the new workload. In case SW doesn't want to save the state of the preempted context, it should at the minimum program RS_PREEMPT_STATUS to 0x0 so that the register status doesn't interfere with the new workloads.

### DWord Bit Description

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:3</td>
<td><strong>Head Pointer Address</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: GraphicsAddress[31:3]</td>
</tr>
</tbody>
</table>

**Description**

This register represents the GFX address offset where execution should continue in the ring buffer following execution of a Preemptable Command. Refer to the Preemption section for the list of preemptable commands supported in ring buffer mode of scheduling.
### UH PTR - Pending Head Pointer Register

<table>
<thead>
<tr>
<th>2:1</th>
<th>Reserved</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Format: MBZ</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>0</th>
<th>Head Pointer Valid</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>This bit is set by the software to request a pre-emption.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>It is reset by hardware when a Preemptable command is parsed by the command streamer. The hardware uses the head pointer programmed in this register at the time the reset is generated. Refer to the Preemption section for the list of preemptable commands supported in ring buffer mode of scheduling.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This bit is treated as set by command streamer only when arbitration is not disabled using MI_ARB_ON_OFF command. Preemption will not occur on MI_ARB_CHEK command when UH PTR is valid if the arbitration is disabled using MI_ARB_ON_OFF command.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>InValid</td>
<td>No valid updated head pointer register, resume execution at the current location in the ring buffer</td>
</tr>
<tr>
<td>1</td>
<td>Valid</td>
<td>Indicates that there is an updated head pointer programmed in this register</td>
</tr>
</tbody>
</table>
## Plink G2H Spare

### SPAREG2H - Plink G2H Spare

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0A194h-0A197h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:9</td>
<td>Reserved</td>
<td>RO</td>
</tr>
<tr>
<td>8:0</td>
<td></td>
<td>Plink_G2H_Spare</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>These are spares for ECO use.</td>
<td></td>
</tr>
</tbody>
</table>
## PM_PWR_CLK_STATE

**Register Space:** MMIO: 0/2/0  
**Project:** CHV, BSW  
**Source:** PRM  
**Default Value:** 0x00008288  
**Size (in bits):** 32  
**Address:** 0A200h  

### PM Power Clock State Request

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31</td>
<td><strong>Enable</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Power Clock State Enable:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 : No specific power state set, no message/wait with PMunit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 : CSunit sends the contents of this register to PMunit each time it is written, Send contents of this register to PMunit, wait for Ack.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>When CS writes to A200, requesting new set of resources:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Actual EUs used = Async_EU if EUmin &lt; Async_EU &lt; EUmax</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Actual EUs used = EUmax if Async_EU &gt; equal to EUmax</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Actual EUs used = EUmin if EUmin &gt; equal to Async_EU</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Actual SSs used = Async_SS if SScountEn=0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Actual SSs used = SScount if SScountEn=1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>After new resources are set by GPMunit, GPM replies to CS by writing 0001_0001 to 0x00_300c</td>
</tr>
<tr>
<td>30:19</td>
<td></td>
<td><strong>RSVD</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved (CSunit implements full 32b storage)</td>
</tr>
<tr>
<td>18</td>
<td></td>
<td><strong>SCountEn</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td>17:15</td>
<td></td>
<td><strong>SliceCount</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 001b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

### Programming Notes

<table>
<thead>
<tr>
<th>Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHV, BSW</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHV, BSW</td>
</tr>
</tbody>
</table>
## PM_PWR_CLK_STATE - PM_PWR_CLK_STATE

<table>
<thead>
<tr>
<th>Address</th>
<th>Register</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>14:13</td>
<td>RSVD</td>
<td>RO</td>
<td>Reserved (CSunit implements full 32b storage)</td>
</tr>
<tr>
<td>12</td>
<td>Spare</td>
<td>R/W</td>
<td>Spare bit for CHV, BSW</td>
</tr>
<tr>
<td>11</td>
<td>SSCountEn</td>
<td>R/W</td>
<td>Enable Subslice Count Request</td>
</tr>
<tr>
<td>10:8</td>
<td>SScount</td>
<td>R/W</td>
<td>Number of subslices to power:</td>
</tr>
<tr>
<td>7:4</td>
<td>EUmax</td>
<td>R/W</td>
<td>Maximum number of EUs to power (per subslice if multiple subslices enabled).</td>
</tr>
<tr>
<td>3:0</td>
<td>EUmin</td>
<td>R/W</td>
<td>Minimum number of EUs to power (per subslice if multiple subslices enabled):</td>
</tr>
</tbody>
</table>

### Programming Notes

<table>
<thead>
<tr>
<th></th>
<th>Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>Must be zero (MBZ).</td>
<td>CHV, BSW</td>
</tr>
</tbody>
</table>

### Notes

- **EUmin and MaxEU need to be even and that odd numbers are illegal**
- **When both subslices are enabled**, the value of EUmax should be equal to EUmin.

### Example

- **EUmax and EUmin** should be set to the same value when both subslices are enabled (e.g., 0x18_2168[11:10] = 2'b00).
## PM_PWR_CLK_STATE - PM_PWR_CLK_STATE

<table>
<thead>
<tr>
<th>Config</th>
<th>Disabled Column</th>
<th>FUSE_GT_EU_DISABLE - 0x18_2168[31:16]</th>
<th>Desired Config</th>
<th>Actual/Driver Config</th>
</tr>
</thead>
<tbody>
<tr>
<td>2x8</td>
<td>none</td>
<td>0x0000</td>
<td>2x8</td>
<td>2x8</td>
</tr>
<tr>
<td>2x8</td>
<td>none</td>
<td>0x0000</td>
<td>2x6</td>
<td>2x6</td>
</tr>
<tr>
<td>2x8</td>
<td>none</td>
<td>0x0000</td>
<td>2x4</td>
<td>2x4</td>
</tr>
<tr>
<td>2x8</td>
<td>none</td>
<td>0x0000</td>
<td>2x2</td>
<td>2x2</td>
</tr>
<tr>
<td>2x6</td>
<td>1</td>
<td>0x1111</td>
<td>2x8</td>
<td>NA</td>
</tr>
<tr>
<td>2x6</td>
<td>1</td>
<td>0x1111</td>
<td>2x6</td>
<td>2x8</td>
</tr>
<tr>
<td>2x6</td>
<td>1</td>
<td>0x1111</td>
<td>2x4</td>
<td>2x6</td>
</tr>
<tr>
<td>2x6</td>
<td>1</td>
<td>0x1111</td>
<td>2x2</td>
<td>2x4</td>
</tr>
<tr>
<td>2x6</td>
<td>2</td>
<td>0x2222</td>
<td>2x8</td>
<td>NA</td>
</tr>
<tr>
<td>2x6</td>
<td>2</td>
<td>0x2222</td>
<td>2x6</td>
<td>2x8</td>
</tr>
<tr>
<td>2x6</td>
<td>2</td>
<td>0x2222</td>
<td>2x4</td>
<td>2x6</td>
</tr>
<tr>
<td>2x6</td>
<td>2</td>
<td>0x2222</td>
<td>2x2</td>
<td>2x2</td>
</tr>
<tr>
<td>2x6</td>
<td>3</td>
<td>0x4444</td>
<td>2x8</td>
<td>NA</td>
</tr>
<tr>
<td>2x6</td>
<td>3</td>
<td>0x4444</td>
<td>2x6</td>
<td>2x8</td>
</tr>
<tr>
<td>2x6</td>
<td>3</td>
<td>0x4444</td>
<td>2x4</td>
<td>2x4</td>
</tr>
<tr>
<td>2x6</td>
<td>3</td>
<td>0x4444</td>
<td>2x2</td>
<td>2x2</td>
</tr>
<tr>
<td>2x6</td>
<td>4</td>
<td>0x8888</td>
<td>2x8</td>
<td>NA</td>
</tr>
<tr>
<td>2x6</td>
<td>4</td>
<td>0x8888</td>
<td>2x6</td>
<td>2x6</td>
</tr>
<tr>
<td>2x6</td>
<td>4</td>
<td>0x8888</td>
<td>2x4</td>
<td>2x4</td>
</tr>
<tr>
<td>2x6</td>
<td>4</td>
<td>0x8888</td>
<td>2x2</td>
<td>2x2</td>
</tr>
<tr>
<td>2x5</td>
<td>1+EU3</td>
<td>0x1919</td>
<td>2x8</td>
<td>NA</td>
</tr>
<tr>
<td>2x5</td>
<td>1+EU3</td>
<td>0x1919</td>
<td>2x5</td>
<td>2x8</td>
</tr>
<tr>
<td>2x5</td>
<td>1+EU3</td>
<td>0x1919</td>
<td>2x4</td>
<td>2x6</td>
</tr>
<tr>
<td>2x5</td>
<td>1+EU3</td>
<td>0x1919</td>
<td>2x2</td>
<td>2x4</td>
</tr>
<tr>
<td>2x5</td>
<td>2+EU3</td>
<td>0x2A2A</td>
<td>2x8</td>
<td>NA</td>
</tr>
<tr>
<td>2x5</td>
<td>2+EU3</td>
<td>0x2A2A</td>
<td>2x5</td>
<td>2x8</td>
</tr>
<tr>
<td>2x5</td>
<td>2+EU3</td>
<td>0x2A2A</td>
<td>2x4</td>
<td>2x6</td>
</tr>
<tr>
<td>2x5</td>
<td>2+EU3</td>
<td>0x2A2A</td>
<td>2x2</td>
<td>2x2</td>
</tr>
<tr>
<td>2x5</td>
<td>3+EU3</td>
<td>0x4C4C</td>
<td>2x8</td>
<td>NA</td>
</tr>
<tr>
<td>-----</td>
<td>-------</td>
<td>--------</td>
<td>------</td>
<td>------</td>
</tr>
<tr>
<td>2x5</td>
<td>3+EU3</td>
<td>0x4C4C</td>
<td>2x5</td>
<td>2x8</td>
</tr>
<tr>
<td>2x5</td>
<td>3+EU3</td>
<td>0x4C4C</td>
<td>2x4</td>
<td>2x4</td>
</tr>
<tr>
<td>2x5</td>
<td>3+EU3</td>
<td>0x4C4C</td>
<td>2x2</td>
<td>2x2</td>
</tr>
<tr>
<td></td>
<td>4+EU2</td>
<td>0x8C8C</td>
<td>2x8</td>
<td>NA</td>
</tr>
<tr>
<td>2x5</td>
<td>4+EU2</td>
<td>0x8C8C</td>
<td>2x5</td>
<td>2x6</td>
</tr>
<tr>
<td>2x5</td>
<td>4+EU2</td>
<td>0x8C8C</td>
<td>2x4</td>
<td>2x4</td>
</tr>
<tr>
<td>2x5</td>
<td>4+EU2</td>
<td>0x8C8C</td>
<td>2x2</td>
<td>2x2</td>
</tr>
</tbody>
</table>
## PMCAPID - PMCAPID

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:27</td>
<td>PME_SUPPORT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PMES The graphics controller does not generate PME.</td>
</tr>
<tr>
<td>26</td>
<td>D2_SUPPORT</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D2S: D2 power management state is not supported.</td>
</tr>
<tr>
<td>25</td>
<td>D1_SUPPORT</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D1S: D1 power management state is not supported.</td>
</tr>
<tr>
<td>24:22</td>
<td>RESERVED</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td>21</td>
<td>DEVICE_SPECIFIC_INITIALIZATION</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 1b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Hardwired to 1 to indicate that special initialization of the graphics controller is required before generic class device driver is to use it.</td>
</tr>
<tr>
<td>20:19</td>
<td>RESERVED</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00b</td>
</tr>
</tbody>
</table>
## PMCAPID - PMCAPID

<table>
<thead>
<tr>
<th>Access</th>
<th>RO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reserved</td>
<td></td>
</tr>
</tbody>
</table>

### 18:16 VERSION

<table>
<thead>
<tr>
<th>Default Value:</th>
<th>010b</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access:</td>
<td>RO</td>
</tr>
</tbody>
</table>

- Version compliance with revision 1.1 of PCI Power management spec.

### 15:8 NEXT_POINTER

<table>
<thead>
<tr>
<th>Default Value:</th>
<th>90h</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access:</td>
<td>R/W Once</td>
</tr>
</tbody>
</table>

- Indicates the next item in the capabilities list (90 = MSI)
- This register should be programmed by BIOS during boot-up. Once written, this register becomes Read Only. This register can only be cleared by a Reset.
- Write once allowing changing of the capabilities list.

### 7:0 CAPABILITIES_ID

<table>
<thead>
<tr>
<th>Default Value:</th>
<th>01h</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access:</td>
<td>RO</td>
</tr>
</tbody>
</table>

CAPID: SIG defines this ID is 01h for power management.
## PMCS - PMCS

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>PCI: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>000D4h</td>
</tr>
</tbody>
</table>

Power Management Control/Status. Driver does not use this register. SBIOS does not use this register.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:2</td>
<td>RESERVED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>1:0</td>
<td></td>
<td>POWER_STATE_PS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

This field indicates the current power state of the IGD and can be used to set the IGD into a new power state. If software attempts to write an unsupported state to this field, write operation must complete normally on the bus, but the data is discarded and no state change occurs. On a transition from D3 to D0 the graphics controller is optionally reset to initial values. Behavior of the graphics controller in supported states is detailed in the power management section of the PRM. Bits[1:0] Power state 00: D0 Default 01: D1 Not Supported 10: D2 Not Supported 11: D3
**POWER_WELL_SS0_SIG1**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x02020242</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0A720h</td>
</tr>
</tbody>
</table>

Subwell signals to be driven out to subslice0 wells: Subwells 1-4.
This register is modified by HW based on sub-power domain configuration and SW should not be writing to this register.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value:</th>
<th>Access:</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:30</td>
<td><strong>Reserved</strong></td>
<td>00b</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Reserved</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>29</td>
<td>ss0_eu210_rst_b</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Subwell control signal 'rst_b' for subwell 'ss0_eu210'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>28</td>
<td>ss0_eu210_fwenb</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Subwell control signal 'fwenb' for subwell 'ss0_eu210'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>27</td>
<td>ss0_eu210_pwrok</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Subwell control signal 'pwrok' for subwell 'ss0_eu210'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>26</td>
<td>ss0_eu210_asynchrst_b</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Subwell control signal 'asynchrst_b' for subwell 'ss0_eu210'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>25</td>
<td>ss0_eu210_pgenb</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### POWER_WELL_SS0_SIG1 - POWER_WELL_SS0_SIG1

<table>
<thead>
<tr>
<th>Offset</th>
<th>Field Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td>ss0_eu210_clocken</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>Default Value:</td>
<td>0b</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Reserved</td>
<td>00b</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td>Subwell control signal 'clocken' for subwell 'ss0_eu210'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23:22</td>
<td>Reserved</td>
<td>00b</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td>Default Value:</td>
<td>00b</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>RO</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Reserved</td>
<td>00b</td>
<td>RO</td>
</tr>
<tr>
<td>21</td>
<td>ss0_eu19_rst_b</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>Default Value:</td>
<td>0b</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Subwell control signal 'rst_b' for subwell 'ss0_eu19'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>ss0_eu19_fwenb</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>Default Value:</td>
<td>0b</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Subwell control signal 'fwenb' for subwell 'ss0_eu19'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>ss0_eu19_pwrok</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>Default Value:</td>
<td>0b</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Subwell control signal 'pwrok' for subwell 'ss0_eu19'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>ss0_eu19_asyncrst_b</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>Default Value:</td>
<td>0b</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Subwell control signal 'asyncrst_b' for subwell 'ss0_eu19'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>ss0_eu19_pgenb</td>
<td>1b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>Default Value:</td>
<td>1b</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Subwell control signal 'pgenb' for subwell 'ss0_eu210'</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Default Value**: Value set at power-on or reset condition.
- **Access**: Read/Write (R/W) or Read Only (RO).
## POWER_WELL_SS0_SIG1 - POWER_WELL_SS0_SIG1

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>ss0_eu19_clocken</td>
<td>0b</td>
<td>R/W</td>
<td>Subwell control signal 'clocken' for subwell 'ss0_eu19'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15:14</td>
<td>Reserved</td>
<td>00b</td>
<td>RO</td>
<td>Reserved</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>ss0_eu08_rst_b</td>
<td>0b</td>
<td>R/W</td>
<td>Subwell control signal 'rst_b' for subwell 'ss0_eu08'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>ss0_eu08_fwenb</td>
<td>0b</td>
<td>R/W</td>
<td>Subwell control signal 'fwenb' for subwell 'ss0_eu08'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>ss0_eu08_pwrok</td>
<td>0b</td>
<td>R/W</td>
<td>Subwell control signal 'pwrok' for subwell 'ss0_eu08'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>ss0_eu08_asyncrst_b</td>
<td>0b</td>
<td>R/W</td>
<td>Subwell control signal 'asyncrst_b' for subwell 'ss0_eu08'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>ss0_eu08_pgenb</td>
<td>1b</td>
<td>R/W</td>
<td>Subwell control signal 'pgenb' for subwell 'ss0_eu08'</td>
<td>0 = sub-well is powered on. 1 = sub-well is powered off.</td>
<td></td>
</tr>
</tbody>
</table>

Subwell control signal 'pgenb' for subwell 'ss0_eu19'
0 = sub-well is powered on.
1 = sub-well is powered off.
<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>POWER_WELL_SS0_SIG1 - POWER_WELL_SS0_SIG1</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>8</strong></td>
<td>ss0_eu08_clocken</td>
<td></td>
</tr>
<tr>
<td>Default Value:</td>
<td>0b</td>
<td></td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>Subwell control signal 'clocken' for subwell 'ss0_eu08'</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>7</strong></td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>Default Value:</td>
<td>0b</td>
<td></td>
</tr>
<tr>
<td>Access:</td>
<td>RO</td>
<td></td>
</tr>
<tr>
<td>Reserved</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>6</strong></td>
<td>ss0_pgenb_sig1</td>
<td></td>
</tr>
<tr>
<td>Default Value:</td>
<td>1b</td>
<td></td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>Subwell control signal 'pgenb' for subwell 'ss0_ss0'</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>5</strong></td>
<td>ss0_rst_b</td>
<td></td>
</tr>
<tr>
<td>Default Value:</td>
<td>0b</td>
<td></td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>Subwell control signal 'rst_b' for subwell 'ss0_ss0'</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>4</strong></td>
<td>ss0_fwenb</td>
<td></td>
</tr>
<tr>
<td>Default Value:</td>
<td>0b</td>
<td></td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>Subwell control signal 'fwenb' for subwell 'ss0_ss0'</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>3</strong></td>
<td>ss0_pwrok</td>
<td></td>
</tr>
<tr>
<td>Default Value:</td>
<td>0b</td>
<td></td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>Subwell control signal 'pwrok' for subwell 'ss0_ss0'</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>2</strong></td>
<td>ss0_asyncrst_b</td>
<td></td>
</tr>
<tr>
<td>Default Value:</td>
<td>0b</td>
<td></td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>Subwell control signal 'asyncrst_b' for subwell 'ss0_ss0'</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>1</strong></td>
<td>ss0_pgenb_sig0</td>
<td></td>
</tr>
<tr>
<td>Default Value:</td>
<td>1b</td>
<td></td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
<td></td>
</tr>
</tbody>
</table>
### POWER_WELL_SS0_SIG1 - POWER_WELL_SS0_SIG1

<table>
<thead>
<tr>
<th>0</th>
<th>ss0_clocken</th>
</tr>
</thead>
</table>
|   | Subwell control signal 'pgenb' for subwell 'ss0_ss0'  
|   | 0 = sub-well is powered on.  
|   | 1 = sub-well is powered off.  
|   | Default Value: 0b  
|   | Access: R/W  
|   | Subwell control signal 'clocken' for subwell 'ss0_ss0' |
## POWER_WELL_SS0_SIG2

### Register Space:
- MMIO: 0/2/0

### Project:
- CHV, BSW

### Source:
- PRM

### Default Value:
- 0x00000002

### Size (in bits):
- 32

### Address:
- 0A724h

Subwell signals to be driven out to subslice0 wells: subwell 5.
This register is modified by HW based on sub-power domain configuration and SW should not be writing to this register.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:6</td>
<td><strong>Reserved</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value:</td>
<td>00000000h</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td><strong>ss0_eu311_rst_b</strong></td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Subwell control signal 'rst_b' for subwell 'ss0_eu311'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td><strong>ss0_eu311_fwenb</strong></td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Subwell control signal 'fwenb' for subwell 'ss0_eu311'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td><strong>ss0_eu311_pwrok</strong></td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Subwell control signal 'pwrok' for subwell 'ss0_eu311'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td><strong>ss0_eu311_asyncrst_b</strong></td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Subwell control signal 'asyncrst_b' for subwell 'ss0_eu311'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td><strong>ss0_eu311_pgenb</strong></td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Subwell control signal 'pgenb' for subwell 'ss0_eu311'</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### POWER_WELL_SS0_SIG2 - POWER_WELL_SS0_SIG2

<table>
<thead>
<tr>
<th></th>
<th>Default Value:</th>
<th>Access:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1b</td>
<td>R/W</td>
</tr>
</tbody>
</table>

Subwell control signal 'pgenb' for subwell 'ss0_eu311'

0 = sub-well is powered on.
1 = sub-well is powered off.

<table>
<thead>
<tr>
<th>0</th>
<th>ss0_eu311_clocken</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Default Value:</td>
</tr>
<tr>
<td></td>
<td>0b</td>
</tr>
</tbody>
</table>

Subwell control signal 'clocken' for subwell 'ss0_eu311'
## POWER_WELL_SS1_SIG1 - POWER_WELL_SS1_SIG1

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:30</td>
<td>Reserved</td>
<td>00b</td>
<td>RO</td>
</tr>
<tr>
<td>29</td>
<td>ss1_eu210_rst_b</td>
<td>Subwell control signal 'rst_b' for subwell 'ss1_eu210'</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td>28</td>
<td>ss1_eu210_fwenb</td>
<td>Subwell control signal 'fwenb' for subwell 'ss1_eu210'</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td>27</td>
<td>ss1_eu210_pwrok</td>
<td>Subwell control signal 'pwrok' for subwell 'ss1_eu210'</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td>26</td>
<td>ss1_eu210_asyncrst_b</td>
<td>Subwell control signal 'asyncrst_b' for subwell 'ss1_eu210'</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td>25</td>
<td>ss1_eu210_pgenb</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Subwell signals to be driven out to subslice1 wells: Subwells 1-4.
This register is modified by HW based on sub-power domain configuration and SW should not be writing to this register.
## POWER_WELL_SS1_SIG1 - POWER_WELL_SS1_SIG1

<table>
<thead>
<tr>
<th>Address</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td>ss1_eu210_clocken</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>Subwell control signal 'clocken' for subwell 'ss1_eu210'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23:22</td>
<td>Reserved</td>
<td>00b</td>
<td>RO</td>
</tr>
<tr>
<td>21</td>
<td>ss1_eu19_rst_b</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>Subwell control signal 'rst_b' for subwell 'ss1_eu19'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>ss1_eu19_fwenb</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>Subwell control signal 'fwenb' for subwell 'ss1_eu19'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>ss1_eu19_pwrok</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>Subwell control signal 'pwrok' for subwell 'ss1_eu19'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>ss1_eu19_asyncrst_b</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>Subwell control signal 'asyncrst_b' for subwell 'ss1_eu19'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>ss1_eu19_pgenb</td>
<td>1b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>Subwell control signal 'pgenb' for subwell 'ss1_eu210'</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Default Value:**
  - 1b
  - 0b
  - 00b

- **Access:**
  - R/W
  - RO
## POWER_WELL_SS1_SIG1 - POWER_WELL_SS1_SIG1

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>Subwell control signal 'pgenb' for subwell 'ss1_eu19'</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 = sub-well is powered on.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 = sub-well is powered off.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15:14</td>
<td>Reserved</td>
<td>00b</td>
<td>RO</td>
</tr>
<tr>
<td>13</td>
<td>Subwell control signal 'clocken' for subwell 'ss1_eu19'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Subwell control signal 'fwenb' for subwell 'ss1_eu08'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Subwell control signal 'pwrok' for subwell 'ss1_eu08'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Subwell control signal 'asyncrst_b' for subwell 'ss1_eu08'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Subwell control signal 'pgenb' for subwell 'ss1_eu08'</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 = sub-well is powered on.</td>
<td>1b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>1 = sub-well is powered off.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Subwell control signal 'pgenb' for subwell 'ss1_eu19'

Subwell control signal 'clocken' for subwell 'ss1_eu19'

Subwell control signal 'fwenb' for subwell 'ss1_eu08'

Subwell control signal 'pwrok' for subwell 'ss1_eu08'

Subwell control signal 'asyncrst_b' for subwell 'ss1_eu08'

Subwell control signal 'pgenb' for subwell 'ss1_eu08'

0 = sub-well is powered on.

1 = sub-well is powered off.
### POWER_WELL_SS1_SIG1 - POWER_WELL_SS1_SIG1

<table>
<thead>
<tr>
<th>Bit</th>
<th>Register</th>
<th>Default Value</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>ss1_eu08_clocken</td>
<td>0b</td>
<td>R/W</td>
<td>Subwell control signal 'clocken' for subwell 'ss1_eu08'</td>
</tr>
<tr>
<td>7</td>
<td>Reserved</td>
<td>0b</td>
<td>RO</td>
<td>Reserved</td>
</tr>
<tr>
<td>6</td>
<td>ss1_pgenb_sig1</td>
<td>1b</td>
<td>R/W</td>
<td>Subwell control signal 'pgenb' for subwell 'ss1_ss1'</td>
</tr>
<tr>
<td>5</td>
<td>ss1_rst_b</td>
<td>0b</td>
<td>R/W</td>
<td>Subwell control signal 'rst_b ' for subwell 'ss1_ss1'</td>
</tr>
<tr>
<td>4</td>
<td>ss1_fwenb</td>
<td>0b</td>
<td>R/W</td>
<td>Subwell control signal 'fwenb' for subwell 'ss1_ss1'</td>
</tr>
<tr>
<td>3</td>
<td>ss1_pwrok</td>
<td>0b</td>
<td>R/W</td>
<td>Subwell control signal 'pwrok' for subwell 'ss1_ss1'</td>
</tr>
<tr>
<td>2</td>
<td>ss1_asyncrst_b</td>
<td>0b</td>
<td>R/W</td>
<td>Subwell control signal 'asyncrst_b' for subwell 'ss1_ss1'</td>
</tr>
<tr>
<td>1</td>
<td>ss1_pgenb_sig0</td>
<td>1b</td>
<td>R/W</td>
<td>Subwell control signal 'pgenb' for subwell 'ss1_ss1'</td>
</tr>
</tbody>
</table>
## POWER_WELL_SS1_SIG1 - POWER_WELL_SS1_SIG1

<table>
<thead>
<tr>
<th></th>
<th>Subwell control signal 'pgenb' for subwell 'ss1_ss1'</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0 = sub-well is powered on.</td>
</tr>
<tr>
<td></td>
<td>1 = sub-well is powered off.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>0</th>
<th>ss1_clocken</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td>Subwell control signal 'clocken' for subwell 'ss1_ss1'</td>
</tr>
</tbody>
</table>
# Command Reference: Registers

## POWER_WELL_SS1_SIG2

### Register Space:
- **MMIO:** 0/2/0

### Project:
- CHV, BSW

### Source:
- PRM

### Default Value:
- 0x00000002

### Size (in bits):
- 32

### Address:
- 0A72Ch

Subwell signals to be driven out to subslice1 wells: subwell 5.
This register is modified by HW based on sub-power domain configuration and SW should not be writing to this register.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:6</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td>5</td>
<td>ss1_eu311_rst_b</td>
<td>Subwell control signal 'rst_b' for subwell 'ss1_eu311'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td>4</td>
<td>ss1_eu311_fwenb</td>
<td>Subwell control signal 'fwenb' for subwell 'ss1_eu311'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td>3</td>
<td>ss1_eu311_pwrok</td>
<td>Subwell control signal 'pwrok' for subwell 'ss1_eu311'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td>2</td>
<td>ss1_eu311_asyncrst_b</td>
<td>Subwell control signal 'asyncrst_b' for subwell 'ss1_eu311'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td><strong>POWER_WELL_SS1_SIG2</strong> - <strong>POWER_WELL_SS1_SIG2</strong></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>--------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td><strong>ss1_eu311_pgenb</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default Value: 1b</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Subwell control signal 'pgenb' for subwell 'ss1_eu311'</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 = sub-well is powered on.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 = sub-well is powered off.</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td><strong>ss1_eu311_clocken</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default Value: 0b</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Subwell control signal 'clocken' for subwell 'ss1_eu311'</td>
<td></td>
</tr>
</tbody>
</table>
## Power Context Save

### PWRCTXSAVE - Power Context Save

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td>Mask Bits</td>
<td>0000h</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value:</td>
<td>0000h</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access:</td>
<td>RO</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Extra Bits 15</td>
<td>Default Value:</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>Extra Bits for future use.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Extra Bits 14</td>
<td>Default Value:</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>Extra Bits for future use.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Extra Bits 13</td>
<td>Default Value:</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>Extra Bits for future use.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Extra Bits 12</td>
<td>Default Value:</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>Extra Bits for future use.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Extra Bits 11</td>
<td>Default Value:</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>Extra Bits for future use.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Extra Bits 10</td>
<td>Default Value:</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>Extra Bits for future use.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### PWRCTXSAVE - Power Context Save

<table>
<thead>
<tr>
<th>9</th>
<th><strong>Power Context Save Request</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Value:</td>
<td>0b</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
</tbody>
</table>

Power Context Save.
Bit[9]. Power Context Save Request.
1'b0: Power context save is not being requested (default).
1'b1: Power context save is being requested.
Unit needs to self-clear this bit upon sampling.
This bit is self clear.

<table>
<thead>
<tr>
<th>8:0</th>
<th><strong>Power Context Save Quad Word Credits</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Value:</td>
<td>000000000b</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
</tbody>
</table>

Power Context Save.
Bits[8:0]. QWord Credits for Power Context Save Request.
An initial length packet is required per power context save session, but that packet does not consume a credit. See protocol description for more details.
Minimum Credits = 1: Unit may send 1 QWord pair.
Maximum Credits = 511: Unit may send 511 QWord pairs.
A QWord pair is defined as a 32-bit register address and the corresponding 32-bits of register data.
Only valid with PWRCTX_SAVE_REQ (Bit9).
### Power context Save Register for LPFC

**LPCSR - Power context Save Register for LPFC**

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:10</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO Reserved.</td>
</tr>
<tr>
<td>9:0</td>
<td></td>
<td><strong>Power context save register command</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Hardware Clear</td>
</tr>
</tbody>
</table>

- **Bit[9]. Power Context Save Request.**
- 1'b0: Power context save is not being requested (default).
- 1'b1: Power context save is being requested.
- Unit needs to self-clear this bit upon sampling.
- **Bits[8:0]. QWord Credits for Power Context Save Request.**
  - Minimum Credits = 1: Unit may send 1 QWord pair (enough for first LRI at least).
  - Maximum Credits = 511: Unit may send 511 QWord pairs.
  - A QWord pair is defined as a 32-bit register address and the corresponding 32-bits of register data. Note that the LRI header and END commands are 64-bits each (32-bit command followed by 32-bit NOOP) and consume one QWord credit.
  - Only valid with PWRCTX_SAVE_REQ (Bit9).
# Power Context Save request

<table>
<thead>
<tr>
<th><strong>PCTXSAVEREQ - Power Context Save request</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 08110h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>DWORD</strong></th>
<th><strong>Bit</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>0</strong></td>
<td>31:16</td>
<td><strong>Message Mask</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Access:</strong></td>
<td><strong>RO</strong></td>
</tr>
<tr>
<td></td>
<td>Message Mask bots for lower 16 bits</td>
<td></td>
</tr>
<tr>
<td><strong>15:10</strong></td>
<td><strong>Reserved</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Access:</strong></td>
<td><strong>RO</strong></td>
</tr>
<tr>
<td></td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td><strong>9</strong></td>
<td><strong>Power context save req</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Access:</strong></td>
<td><strong>R/W Set</strong></td>
</tr>
<tr>
<td></td>
<td>Power Context Save Request</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1'b0 : Power context save is not being requested (default)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1'b1 : Power context save is being requested</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CPUnit self-clears this bit upon sampling.</td>
<td></td>
</tr>
<tr>
<td><strong>8:0</strong></td>
<td><strong>Power Context Save request credit count</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Access:</strong></td>
<td><strong>R/W</strong></td>
</tr>
<tr>
<td></td>
<td>QWord Credits for Power Context Save Request</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Minimum Credits = 1 : Unit may send 1 QWord pair (enough for first LRI at least)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Maximum Credits = 511 : Unit may send 511 QWord pairs</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A QWord pair is defined as a 32-bit register address and the corresponding 32-bits of register data. Note that the LRI header and END commands are 64-bits each (32-bit command followed by 32-bit NOOP) and will consume one QWord credit.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Only valid with PWRCTX_SAVE_REQ (Bit9).</td>
<td></td>
</tr>
</tbody>
</table>
## POWERDOWN_STATE

**Register Space:** MMIO: 0/2/0  
**Project:** CHV, BSW  
**Source:** PRM  
**Default Value:** 0x66654321  
**Size (in bits):** 32  
**Address:** 0A710h  

**Subwell Power down state description.**

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:28</td>
<td><strong>pmcr_powerdown_state8</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0110b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The 8th state/signal that needs to be driven</td>
</tr>
<tr>
<td></td>
<td>27:24</td>
<td><strong>pmcr_powerdown_state7</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0110b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The 7th state/signal that needs to be driven</td>
</tr>
<tr>
<td></td>
<td>23:20</td>
<td><strong>pmcr_powerdown_state6</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0110b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The 6th state/signal that needs to be driven</td>
</tr>
<tr>
<td></td>
<td>19:16</td>
<td><strong>pmcr_powerdown_state5</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0101b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The 5th state/signal that needs to be driven</td>
</tr>
<tr>
<td></td>
<td>15:12</td>
<td><strong>pmcr_powerdown_state4</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0100b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The 4th state/signal that needs to be driven</td>
</tr>
<tr>
<td></td>
<td>11:8</td>
<td><strong>pmcr_powerdown_state3</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0011b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>
### POWERDOWN_STATE - POWERDOWN_STATE

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7:4 pmcr_powerdown_state2</td>
<td>The 3rd state/signal that needs to be driven</td>
</tr>
<tr>
<td>Default Value</td>
<td>0010b</td>
</tr>
<tr>
<td>Access</td>
<td>R/W</td>
</tr>
</tbody>
</table>

The 3rd state/signal that needs to be driven

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3:0 pmcr_powerdown_state1</td>
<td>The 2nd state/signal that needs to be driven</td>
</tr>
<tr>
<td>Default Value</td>
<td>0001b</td>
</tr>
<tr>
<td>Access</td>
<td>R/W</td>
</tr>
</tbody>
</table>

The 2nd state/signal that needs to be driven

This is the very first state and the signal that needs to be driven appropriately

**Encodings:**
- 001 = Reset Sync Reset
- 010 = Reset Firewall
- 011 = Set Power EnB
- 100 = Reset Clock En
- 101 = Reset Async Reset
- 110 = Done
- 111 = Reserved.

**Open:** Do we want to have an extra bit for future support
POWERDOWN_WAIT1

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x01010101</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0A714h</td>
</tr>
</tbody>
</table>

Subwell power down state wait time 1.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><em>pmcr_powerdown_wait_state4_5</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 01h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This is the programmed value as to how long there is a wait from the time signal 4 is driven before signal 5 can be driven</td>
</tr>
<tr>
<td>0</td>
<td>23:16</td>
<td><em>pmcr_powerdown_wait_state3_4</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 01h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This is the programmed value as to how long there is a wait from the time signal 3 is driven before signal 4 can be driven</td>
</tr>
<tr>
<td>0</td>
<td>15:8</td>
<td><em>pmcr_powerdown_wait_state2_3</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 01h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This is the programmed value as to how long there is a wait from the time signal 2 is driven before signal 3 can be driven</td>
</tr>
<tr>
<td>0</td>
<td>7:0</td>
<td><em>pmcr_powerdown_wait_state1_2</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 01h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This is the programmed value as to how long there is a wait from the time signal 1 is driven before signal 2 can be driven. Note: The granularity of these are in 30ns intervals meaning the counters will increment/decrement every 30ns pulse</td>
</tr>
</tbody>
</table>
### POWERDOWN_WAIT2

Subwell power down state wait time 2.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>pmcr_powerdown_interval</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 01h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Stagger between different power gate enables for power down.</td>
</tr>
<tr>
<td></td>
<td>23:16</td>
<td><strong>pmcr_powerdown_wait_state7_8</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 01h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This is the programmed value as to how long there is a wait from the time signal 7 is driven before signal 8 can be driven</td>
</tr>
<tr>
<td></td>
<td>15:8</td>
<td><strong>pmcr_powerdown_wait_state6_7</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 01h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This is the programmed value as to how long there is a wait from the time signal 6 is driven before signal 7 can be driven</td>
</tr>
<tr>
<td></td>
<td>7:0</td>
<td><strong>pmcr_powerdown_wait_state5_6</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 01h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This is the programmed value as to how long there is a wait from the time signal 5 is driven before signal 6 can be driven</td>
</tr>
</tbody>
</table>
# Power Down/Up Control

## PWRDWNUPCTL - Power Down/Up Control

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0A294h-0A297h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:4</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td><strong>Block Signaling Policy</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BlockAll signaling policy register. Applies to signaling sent to Wake FIFO (ie. IA GT FIFO and HW FIFO).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 (default) : BlockAll signal only asserts for CPD.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 : BlockAll signal can assert for CPD and for a timeperiod during RC6 entry.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note : This affects MMIO requests crossing from CZ clock domain to message channel in Gfx clock domain.</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td><strong>Serialize Power Requests</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Determines whether simultaneous requests to REMOVE power are presented to the Punit at the same time or whether the handshake for one must complete before the handshake for the other may start.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0=simultaneous</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1=serial</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td><strong>Powerdown Request Order</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Determines which well is disabled first if both wells are requesting to enter standby simultaneously, if serialized.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0=render well power is removed first</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1=media well power is removed first.</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td><strong>Wake Media First</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>When waking both wells, it determines which well wakes first.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0=wake render first</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1=wake media first.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(It is NOT possible to wake both simultaneously for di/dt reasons)</td>
</tr>
</tbody>
</table>
Power Enable stagger control

<table>
<thead>
<tr>
<th>PWRENSTAGCNTRL - Power Enable stagger control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000001</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 0A70Ch</td>
</tr>
<tr>
<td>Power Enable Stagger Control</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:8</td>
<td><strong>Reserved</strong></td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7:0</td>
<td></td>
<td><strong>pgen_phase_interval</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 01h</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 01h</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>It is the separation between pgen0 and pgen1 on SSx. (units = 30ns)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default value 0x01. (Real silicon settings will be determined by circuit simulation, but expect it to be no higher than 0x21. Nominally, expect it to be 0x0A)</td>
<td></td>
</tr>
</tbody>
</table>
### PMWGICZ - Power Meter Weight for gti_idle_cz

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td><strong>RESERVED</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td>15:0</td>
<td></td>
<td><strong>PMWGICZ</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

This register contains the power meter weight for the idleness in CZ clock domain.

- **Register Space:** MMIO: 0/2/0
- **Project:** CHV, BSW
- **Source:** PRM
- **Default Value:** 0x00000000
- **Size (in bits):** 32
- **Address:** 1300A4h

This 16-bit value is used to indicate how often to accumulate power within the CZ clock domain. When this field is set to zero, CZ clock domain will not contribute to the overall energy count. Otherwise, the value in this register will dictate how often to add CZ power contribution.
### PWRMTR_WT_GTI - Power Meter Weight for gti_idle gs

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0AABCh</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>15:0</td>
<td>Power Meter Weight GTI</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
<td></td>
</tr>
</tbody>
</table>

Power meter weight for gti_idle_gs.
# Power Meter Weight for media/render

<table>
<thead>
<tr>
<th><strong>Register Space</strong></th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Project</strong></td>
<td>CHV, BSW</td>
</tr>
<tr>
<td><strong>Source</strong></td>
<td>PRM</td>
</tr>
<tr>
<td><strong>Default Value</strong></td>
<td>0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits)</strong></td>
<td>32</td>
</tr>
<tr>
<td><strong>Address</strong></td>
<td>0AAB8h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td><strong>Power Meter Weight Render</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Power meter weight for render_idle.</td>
<td></td>
</tr>
<tr>
<td>15:0</td>
<td></td>
<td><strong>Power Meter Weight Media</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Power meter event weight media_idle.</td>
<td></td>
</tr>
</tbody>
</table>

---

*Doc Ref #: IHD-OS-CHV-BSW-Vol 2c-10.15*
### POWERUP_STATE

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:28</td>
<td>pmcr_powerup_state8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0111b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The 8th state/signal that needs to be driven</td>
</tr>
<tr>
<td></td>
<td>27:24</td>
<td>pmcr_powerup_state7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0110b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The 7th state/signal that needs to be driven</td>
</tr>
<tr>
<td></td>
<td>23:20</td>
<td>pmcr_powerup_state6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0101b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The 6th state/signal that needs to be driven</td>
</tr>
<tr>
<td></td>
<td>19:16</td>
<td>pmcr_powerup_state5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0100b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The 5th state/signal that needs to be driven</td>
</tr>
<tr>
<td></td>
<td>15:12</td>
<td>pmcr_powerup_state4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0011b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The 4th state/signal that needs to be driven</td>
</tr>
<tr>
<td></td>
<td>11:8</td>
<td>pmcr_powerup_state3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0010b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>
The 3rd state/signal that needs to be driven

<table>
<thead>
<tr>
<th>7:4</th>
<th>pmcr_powerup_state2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Value:</td>
<td>0001b</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
</tbody>
</table>

The 2nd state/signal that needs to be driven

<table>
<thead>
<tr>
<th>3:0</th>
<th>pmcr_powerup_state1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Value:</td>
<td>0000b</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
</tbody>
</table>

This is the very first state and the signal that needs to be driven appropriately

Encodings:
- 0000 = First Set Clock En
- 0001 = Reset Power EnB
- 0010 = Reset Clock En
- 0011 = Set Firewall
- 0100 = Set Async Reset/ Set Pwrok
- 0101 = Final Set clock en
- 0110 = Set Sync Reset.
- 0111 = Done
- 1xxx = Reserved for future
## POWERUP_WAIT1

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><code>pmcr_powerup_wait_state4_5</code>&lt;br&gt;Default Value: 01h&lt;br&gt;Access: R/W&lt;br&gt;This is the programmed value as to how long there is a wait from the time signal 4 is driven before signal 5 can be driven</td>
</tr>
<tr>
<td>23:16</td>
<td></td>
<td><code>pmcr_powerup_wait_state3_4</code>&lt;br&gt;Default Value: 01h&lt;br&gt;Access: R/W&lt;br&gt;This is the programmed value as to how long there is a wait from the time signal 3 is driven before signal 4 can be driven</td>
</tr>
<tr>
<td>15:8</td>
<td></td>
<td><code>pmcr_powerup_wait_state2_3</code>&lt;br&gt;Default Value: 01h&lt;br&gt;Access: R/W&lt;br&gt;This is the programmed value as to how long there is a wait from the time signal 2 is driven before signal 3 can be driven</td>
</tr>
<tr>
<td>7:0</td>
<td></td>
<td><code>pmcr_powerup_wait_state1_2</code>&lt;br&gt;Default Value: 01h&lt;br&gt;Access: R/W&lt;br&gt;This is the programmed value as to how long there is a wait from the time signal 1 is driven before signal 2 can be driven.&lt;br&gt;Note: The granularity of these are in 30ns intervals meaning the counters will increment or decrement every 30ns pulse unless we are asserting clocken in which case it is on a usync boundary</td>
</tr>
</tbody>
</table>
## POWERUP_WAIT2

### Register Space:
- MMIO: 0/2/0

### Project:
- CHV, BSW

### Source:
- PRM

### Default Value:
- 0x01010101

### Size (in bits):
- 32

### Address:
- 0A708h

Subwell power up state wait time 2.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>pmcr_powerup_interval</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 01h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Stagger between different power gate enables on power up</td>
</tr>
<tr>
<td>23:16</td>
<td></td>
<td><strong>pmcr_powerup_wait_state7_8</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 01h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This is the programmed value as to how long there is a wait from the time signal 7 is driven before signal 8 can be driven</td>
</tr>
<tr>
<td>15:8</td>
<td></td>
<td><strong>pmcr_powerup_wait_state6_7</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 01h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This is the programmed value as to how long there is a wait from the time signal 6 is driven before signal 7 can be driven</td>
</tr>
<tr>
<td>7:0</td>
<td></td>
<td><strong>pmcr_powerup_wait_state5_6</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 01h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This is the programmed value as to how long there is a wait from the time signal 5 is driven before signal 6 can be driven</td>
</tr>
</tbody>
</table>
PPGTT Page Fault Data Registers

<table>
<thead>
<tr>
<th>PP_PFD[0:31] - PPGTT Page Fault Data Registers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: RenderCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Access: RO</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 04580h</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:12</td>
<td><strong>Fault Entry Page Address</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: GraphicsAddress[31:12]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This RO field contains the faulting page address for this Fault Log entry. This field will contain a valid fault address only if the bit in the GTT Page Fault Indication Register corresponding with the address offset of this entry is set.</td>
</tr>
<tr>
<td>11:0</td>
<td><strong>Reserved</strong></td>
<td>Format: MBZ</td>
</tr>
</tbody>
</table>
**MI_PREDICATE_RESULT - Predicate Rendering Data Result**

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:1</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MBZ</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td>MI_PREDICATE_RESULT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This bit is the result of the last MI_PREDICATE.</td>
</tr>
</tbody>
</table>
# Predicate Rendering Data Result 1

<table>
<thead>
<tr>
<th><strong>MI_PREDICATE_RESULT_1</strong> - Predicate Rendering Data Result 1</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 0241Ch-0241Fh</td>
</tr>
<tr>
<td><strong>Name:</strong> Predicate Rendering Data Result 1</td>
</tr>
<tr>
<td><strong>ShortName:</strong> MI_PREDICATE_RESULT_1_RCSUNIT</td>
</tr>
<tr>
<td><strong>Address:</strong> 1241Ch-1241Fh</td>
</tr>
<tr>
<td><strong>Name:</strong> Predicate Rendering Data Result 1</td>
</tr>
<tr>
<td><strong>ShortName:</strong> MI_PREDICATE_RESULT_1_VCSUNIT0</td>
</tr>
<tr>
<td><strong>Address:</strong> 1A41Ch-1A41Fh</td>
</tr>
<tr>
<td><strong>Name:</strong> Predicate Rendering Data Result 1</td>
</tr>
<tr>
<td><strong>ShortName:</strong> MI_PREDICATE_RESULT_1_VECSUNIT</td>
</tr>
<tr>
<td><strong>Address:</strong> 1C41Ch-1C41Fh</td>
</tr>
<tr>
<td><strong>Name:</strong> Predicate Rendering Data Result 1</td>
</tr>
<tr>
<td><strong>ShortName:</strong> MI_PREDICATE_RESULT_1_VCSUNIT1</td>
</tr>
<tr>
<td><strong>Address:</strong> 2241Ch-2241Fh</td>
</tr>
<tr>
<td><strong>Name:</strong> Predicate Rendering Data Result 1</td>
</tr>
<tr>
<td><strong>ShortName:</strong> MI_PREDICATE_RESULT_1_BCSUNIT</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>DWord</strong></th>
<th><strong>Bit</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:1</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td>Format:</td>
<td></td>
<td>MBZ</td>
</tr>
<tr>
<td>0</td>
<td><strong>MI_PREDICATE_RESULT_1</strong></td>
<td></td>
</tr>
</tbody>
</table>

This bit is used to predicate MI_BATCH_BUFFER_START commands in the RCS command stream. Usage Model: MI_MATH command will be used to do some ALU operations over GPR followed by a MI_LOAD_REGISTER_REGISTER to move the result from GPR to MI_PREDICATE_RESULT_1.
## Predicate Rendering Data Result 2

<table>
<thead>
<tr>
<th>MI_PREDICATE_RESULT_2 - Predicate Rendering Data Result 2</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
<td></td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
<td></td>
</tr>
<tr>
<td>Source: PRM</td>
<td></td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
<td></td>
</tr>
<tr>
<td>Access: R/W</td>
<td></td>
</tr>
<tr>
<td>Size (in bits): 32</td>
<td></td>
</tr>
<tr>
<td>Address: 023BCh-023Fh</td>
<td></td>
</tr>
<tr>
<td>Name: Predicate Rendering Data Result 2</td>
<td></td>
</tr>
<tr>
<td>ShortName: MI_PREDICATE_RESULT_2_RCSUNIT</td>
<td></td>
</tr>
<tr>
<td>Valid Projects: CHV, BSW</td>
<td></td>
</tr>
<tr>
<td>Address: 123BCh-123Fh</td>
<td></td>
</tr>
<tr>
<td>Name: Predicate Rendering Data Result 2</td>
<td></td>
</tr>
<tr>
<td>ShortName: MI_PREDICATE_RESULT_2_VCSUNIT0</td>
<td></td>
</tr>
<tr>
<td>Valid Projects: CHV, BSW</td>
<td></td>
</tr>
<tr>
<td>Address: 1A3BCh-1A3BFh</td>
<td></td>
</tr>
<tr>
<td>Name: Predicate Rendering Data Result 2</td>
<td></td>
</tr>
<tr>
<td>ShortName: MI_PREDICATE_RESULT_2_VECSUNIT</td>
<td></td>
</tr>
<tr>
<td>Valid Projects: CHV, BSW</td>
<td></td>
</tr>
<tr>
<td>Address: 1C3BCh-1C3BFh</td>
<td></td>
</tr>
<tr>
<td>Name: Predicate Rendering Data Result 2</td>
<td></td>
</tr>
<tr>
<td>ShortName: MI_PREDICATE_RESULT_2_VCSUNIT1</td>
<td></td>
</tr>
<tr>
<td>Valid Projects: CHV, BSW</td>
<td></td>
</tr>
<tr>
<td>Address: 223BCh-223Fh</td>
<td></td>
</tr>
<tr>
<td>Name: Predicate Rendering Data Result 2</td>
<td></td>
</tr>
<tr>
<td>ShortName: MI_PREDICATE_RESULT_2_BCSUNIT</td>
<td></td>
</tr>
</tbody>
</table>

### DWord 0

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>31:1</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

| Format: | MBZ |

<table>
<thead>
<tr>
<th>MI_PREDICATE_RESULT_2</th>
</tr>
</thead>
</table>

This bit must be loaded with by SW based on GT mode of operation. This register must be loaded appropriately before using MI_SET_PREDICATE command.

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>[Default]</td>
<td>Indicates GT2 mode and lower slice is disabled.</td>
</tr>
<tr>
<td>1h</td>
<td></td>
<td>Indicates GT3 mode and lower slice is enabled.</td>
</tr>
</tbody>
</table>
## MI_PREDICATE_DATA - Predicate Rendering Data Storage

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>RenderCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000, 0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>64</td>
</tr>
<tr>
<td>Address:</td>
<td>02410h-02417h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Valid Projects:</th>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>63:32</td>
<td><strong>MI_PREDICATE_DATA_UDW</strong>  &lt;br&gt;This register is used either as computed value based off the MI_PREDICATE_SRC0 and MI_PREDICATE_SRC1 or a temporary register. See Predicate Rendering section for more details.</td>
</tr>
<tr>
<td></td>
<td>31:0</td>
<td><strong>MI_PREDICATE_DATA_LDW</strong>  &lt;br&gt;This register is used either as computed value based off the MI_PREDICATE_SRC0 and MI_PREDICATE_SRC1 or a temporary register. See Predicate Rendering section for more details.</td>
<td></td>
</tr>
</tbody>
</table>
### Predicate Rendering Temporary Register0

<table>
<thead>
<tr>
<th><strong>MI_PREDICATE_SRC0</strong> - Predicate Rendering Temporary Register0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: RenderCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000, 0x00000000</td>
</tr>
<tr>
<td>Access: R/W</td>
</tr>
<tr>
<td>Size (in bits): 64</td>
</tr>
</tbody>
</table>

| Address: 02400h-02407h |

**Valid Projects:**

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>63:0</td>
<td><strong>MI_PREDICATE_SRC0</strong></td>
</tr>
</tbody>
</table>

This register is a temporary register for Predicate Rendering. See Predicate Rendering section for more details.
## Predicate Rendering Temporary Register1

### MI_PREDICATE_SRC1 - Predicate Rendering Temporary Register1

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space</td>
<td>MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source</td>
<td>RenderCS</td>
</tr>
<tr>
<td>Default Value</td>
<td>0x00000000, 0x00000000</td>
</tr>
<tr>
<td>Access</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits)</td>
<td>64</td>
</tr>
<tr>
<td>Address</td>
<td>02408h-0240Fh</td>
</tr>
<tr>
<td>Valid Projects</td>
<td>CHV, BSW</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>63:0</td>
<td>MI_PREDICATE_SRC1</td>
</tr>
</tbody>
</table>

This register is a temporary register for Predicate Rendering. See Predicate Rendering section for more details.
## Previous Idle/Busy/Avg Count for Freq Down Recommendation

### RPPREVDN - Previous Idle/Busy/Avg Count for Freq Down Recommendation

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0A064h-0A067h</td>
</tr>
</tbody>
</table>

### Description

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td>0</td>
<td>23:0</td>
<td>Previous Busy in Down EI</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Previous Busy in Down EI (PRVBSYTAVG): Reports the busyness at the end of the previous down evaluation interval</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 = 0 usec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 = 1.28 usec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 = 2.56 usec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 = 3.84 usec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FF FFFF = 21.474 sec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>pmcr_previous_ei_down_busy[23:0]</td>
</tr>
</tbody>
</table>
## RPPREVUP - Previous Idle/Busy/Avg Count for Freq Up Recommendation

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0A058h-0A05Bh</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td>Reserved</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>23:0</td>
<td></td>
<td>Previous Busy in UP EI</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reports the busyness at the end of the previous Up evaluation interval</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 = 0 usec</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 = 1.28 usec</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 = 2.56 usec</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 = 3.84 usec</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>FF FFFF = 21.474 sec</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>pmcr_previous_ei_up_busy[23:0]</td>
<td></td>
</tr>
</tbody>
</table>
### IA_PRIMITIVES_COUNT - Primitives Generated By VF

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>RenderCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000, 0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>64</td>
</tr>
<tr>
<td>Trusted Type:</td>
<td>1</td>
</tr>
</tbody>
</table>

**Address:** 02318h  
**Valid Projects:**

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
</table>
| 0     | 63:32 | **IA Primitives Count Report UDW**  
Total number of primitives output by the Vertex Fetch (IA) stage. This count is updated for every primitive output by the VF stage, as long as Statistics Enable is set in VF_STATE (see the Vertex Fetch Chapter in the 3D Volume.) |
| 31:0  |      | **IA Primitives Count Report LDW**  
Total number of primitives output by the Vertex Fetch (IA) stage. This count is updated for every primitive output by the VF stage, as long as Statistics Enable is set in VF_STATE (see the Vertex Fetch Chapter in the 3D Volume.) |
**PRIVATE PAT - Private PAT**

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Private PAT</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000003h</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Description</th>
<th>Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bit[31:8]: Reserved.</td>
<td></td>
</tr>
<tr>
<td>Bit[7]: Reserved.</td>
<td></td>
</tr>
<tr>
<td>Bit[6]: Snoop Required [CHV, BSW Only]</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>1: System agent will snoop the IA cores</td>
<td></td>
</tr>
<tr>
<td>0: System agent will not snoop the IA cores</td>
<td></td>
</tr>
<tr>
<td>Bit[5:4]: (See below.)</td>
<td></td>
</tr>
<tr>
<td>00b: Age is 0.</td>
<td></td>
</tr>
<tr>
<td>01b: Age is 1.</td>
<td></td>
</tr>
<tr>
<td>10b: Age is 2.</td>
<td></td>
</tr>
<tr>
<td>11b: Age is 3.</td>
<td></td>
</tr>
<tr>
<td>Bit[3:2]: (See below.)</td>
<td></td>
</tr>
<tr>
<td>00b: eLLC only.</td>
<td></td>
</tr>
<tr>
<td>01b: LLC only.</td>
<td></td>
</tr>
<tr>
<td>10b: LLC and eLLC allowed.</td>
<td></td>
</tr>
<tr>
<td>11b: L3, LLC, and eLLC are allowed.</td>
<td></td>
</tr>
<tr>
<td>Bit[1:0]: (see below):</td>
<td></td>
</tr>
<tr>
<td>00b: Uncacheable (UC).</td>
<td></td>
</tr>
<tr>
<td>01b: Write Combining (WC).</td>
<td></td>
</tr>
<tr>
<td>10b: Write Through (WT).</td>
<td></td>
</tr>
<tr>
<td>11b: Write Back (WB).</td>
<td></td>
</tr>
</tbody>
</table>
## Private PAT

<table>
<thead>
<tr>
<th>PRV_PAT - Private PAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space:</td>
</tr>
<tr>
<td>Project:</td>
</tr>
<tr>
<td>Source:</td>
</tr>
<tr>
<td>Default Value:</td>
</tr>
<tr>
<td>Size (in bits):</td>
</tr>
<tr>
<td>Address:</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Private PAT</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[31:16]: Reserved.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[15:8]: PPGTT Private PAT.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(See bit[7:0] for definition.)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[7:6]: Reserved.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[5:4]: (See below.)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>00b: Age is 0.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>01b: Age is 1.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10b: Age is 2.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>11b: Age is 3.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[3:2]: (See below.)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>00b: Override to eLLC Only. (This setting overrides the memory_object_control_state via surface state to be eLLC target only.)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>01b: eLLC only.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10b: LLC only.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>11b: eLLC/LLC.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[1:0]: (see below):</td>
</tr>
<tr>
<td></td>
<td></td>
<td>00b: Uncached with fence.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>01b: Write Combining (traditional UC).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10b: Write Through.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>11b: Write Back.</td>
</tr>
</tbody>
</table>
### PS Depth Count

#### PS_DEPTH_COUNT - PS Depth Count

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>RenderCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000, 0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>64</td>
</tr>
<tr>
<td>Trusted Type:</td>
<td>1</td>
</tr>
<tr>
<td>Address:</td>
<td>02350h</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0..1</td>
<td>63:32</td>
<td><strong>Depth Count UDW</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>This register reflects the total number of samples that have passed the depth test (i.e., will be visible). All samples are counted when Statistics Enable is set in the Windower State. See the Windower chapter of the 3D volume for details. Samples that pass the depth test but fail the stencil test will not be counted.</td>
</tr>
<tr>
<td>31:0</td>
<td></td>
<td><strong>Depth Count LDW</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>This register reflects the total number of samples that have passed the depth test (i.e., will be visible). All samples are counted when Statistics Enable is set in the Windower State. See the Windower chapter of the 3D volume for details. Samples that pass the depth test but fail the stencil test will not be counted.</td>
</tr>
</tbody>
</table>
# PS Depth Count for Slice0

<table>
<thead>
<tr>
<th><strong>PS_DEPTH_COUNT_SLICE0 - PS Depth Count for Slice0</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> RenderCS</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000, 0x00000000</td>
</tr>
<tr>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 64</td>
</tr>
<tr>
<td><strong>Address:</strong> 02D8h</td>
</tr>
</tbody>
</table>

This register stores the value of the count of pixels that have passed the depth test in Slice0. This register is part of the context save and restore. This register should not be programmed by SW. Note: In GT3 mode there are two slices Slice0 (Upper Slice) and Slice1 (Lower slice).

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0..1</td>
<td>63:32</td>
<td><strong>Depth Count UDW</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>This register reflects the total number of pixels that have passed the depth test in Slice0 (i.e., will be visible). All pixels are counted when Statistics Enable is set in the Windower State. See the Windower chapter of the 3D volume for details. Pixels that pass the depth test but fail the stencil test will not be counted.</td>
</tr>
<tr>
<td>31:0</td>
<td></td>
<td><strong>Depth Count LDW</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>This register reflects the total number of pixels that have passed the depth test in Slice0 (i.e., will be visible). All pixels are counted when Statistics Enable is set in the Windower State. See the Windower chapter of the 3D volume for details. Pixels that pass the depth test but fail the stencil test will not be counted.</td>
</tr>
</tbody>
</table>
# PS Depth Count for Slice1

<table>
<thead>
<tr>
<th>PS_DEPTH_COUNT_SLICE1 - PS Depth Count for Slice1</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> RenderCS</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000, 0x00000000</td>
</tr>
<tr>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 64</td>
</tr>
<tr>
<td><strong>Trusted Type:</strong> 1</td>
</tr>
<tr>
<td><strong>Address:</strong> 022F8h</td>
</tr>
</tbody>
</table>

This register stores the value of the count of pixels that have passed the depth test in Slice1. This register is part of the context save and restore. This register should not be programmed by SW. Note: In GT3 mode there are two slices Slice0 (Upper Slice) and Slice1 (Lower slice).

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
</table>
| 0..1  | 63:32 | **Depth Count UDW**  
This register reflects the total number of pixels that have passed the depth test inSlice1 (i.e., will be visible). All pixels are counted when Statistics Enable is set in the Windower State. See the Windower chapter of the 3D volume for details. Pixels that pass the depth test but fail the stencil test will not be counted. |
| 31:0 | | **Depth Count LDW**  
This register reflects the total number of pixels that have passed the depth test in Slice1 (i.e., will be visible). All pixels are counted when Statistics Enable is set in the Windower State. See the Windower chapter of the 3D volume for details. Pixels that pass the depth test but fail the stencil test will not be counted. |
## PS Depth Count for Slice2

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>RenderCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000, 0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>64</td>
</tr>
<tr>
<td>Trusted Type:</td>
<td>1</td>
</tr>
</tbody>
</table>

Address: 02450h

This register stores the value of the count of pixels that have passed the depth test in Slice2. This register is part of the render context save and restore. This register should not be programmed by SW.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
</table>
| 0..1  | 63:32 | **Depth Count UDW**  
This register reflects the depth test in slice2 (i.e., will be visible). All pixels are counted when Statistics Enable is set in the Windower State. See the Windower chapter of the 3D volume for details. Pixels that pass the depth test but fail the stencil test will not be counted. |
|       | 31:0  | **Depth Count LDW**  
This register reflects the depth test in slice2 (i.e., will be visible). All pixels are counted when Statistics Enable is set in the Windower State. See the Windower chapter of the 3D volume for details. Pixels that pass the depth test but fail the stencil test will not be counted. |
PS Depth Count for Slice3

<table>
<thead>
<tr>
<th>PS_DEPTH_COUNT_SLICE3 - PS Depth Count for Slice3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: RenderCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000, 0x00000000</td>
</tr>
<tr>
<td>Access: R/W</td>
</tr>
<tr>
<td>Size (in bits): 64</td>
</tr>
<tr>
<td>Trusted Type: 1</td>
</tr>
</tbody>
</table>

Address: 02460h

This register stores the value of the count of pixels that have passed the depth test in Slice3. This register is part of the render context save and restore. This register should not be programmed by SW.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0..1</td>
<td>63:32</td>
<td><strong>Depth Count UDW</strong>&lt;br&gt;This register reflects the depth test in slice3 (i.e., will be visible). All pixels are counted when Statistics Enable is set in the Windower State. See the Windower chapter of the 3D volume for details. Pixels that pass the depth test but fail the stencil test will not be counted.</td>
</tr>
<tr>
<td></td>
<td>31:0</td>
<td><strong>Depth Count LDW</strong>&lt;br&gt;This register reflects the depth test in slice3 (i.e., will be visible). All pixels are counted when Statistics Enable is set in the Windower State. See the Windower chapter of the 3D volume for details. Pixels that pass the depth test but fail the stencil test will not be counted.</td>
</tr>
</tbody>
</table>
# PS Invocation Count

<table>
<thead>
<tr>
<th>PS_INVOCATION_COUNT - PS Invocation Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: RenderCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000, 0x00000000</td>
</tr>
<tr>
<td>Access: R/W</td>
</tr>
<tr>
<td>Size (in bits): 64</td>
</tr>
<tr>
<td>Trusted Type: 1</td>
</tr>
<tr>
<td>Address: 02348h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
</table>
| 0..1  | 63:32 | **PS Invocation Count UDW**  
|       |      | Reflects a count of the total number of pixels (including unlit "helper pixels" within a subspan that need to go through the PS shader to provide 2x2 gradients) that are dispatched to pixel shader invocations while Statistics Enable is set in the Windower. See the Windower chapter of the 3D volume for details. This count will generally be much greater than the actual count of PS threads since a single thread may process up to 32 pixels. |
| 31:0  | **PS Invocation Count LDW**  
|       |      | Reflects a count of the total number of pixels (including unlit "helper pixels" within a subspan that need to go through the PS shader to provide 2x2 gradients) that are dispatched to pixel shader invocations while Statistics Enable is set in the Windower. See the Windower chapter of the 3D volume for details. This count will generally be much greater than the actual count of PS threads since a single thread may process up to 32 pixels. |
PS Invocation Count for Slice0

**PS_INVOCATION_COUNT_SLICE0 - PS Invocation Count for Slice0**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>RenderCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000, 0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>64</td>
</tr>
<tr>
<td>Trusted Type:</td>
<td>1</td>
</tr>
<tr>
<td>Address:</td>
<td>022C8h</td>
</tr>
</tbody>
</table>

This register stores the value of the count of pixels that get shaded in Slice0. This register is part of the context save and restore. This register should not be programmed by SW. Note: In GT3 mode there are two slices Slice0 (Upper Slice) and Slice1 (Lower slice).

**Workaround**

HW reports this count 4X the actual value and therefore SW must divide the count by 4 for correct reporting.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0..1</td>
<td>63:32</td>
<td><strong>PS Invocation Count UDW</strong>&lt;br&gt;Reflects a count of the total number of pixels (including unlit &quot;helper pixels&quot; within a subspan that need to go through the PS shader to provide 2x2 gradients) that are dispatched to pixel shader invocations while Statistics Enable is set in the Windower in Slice0. See the Windower chapter of the 3D volume for details. This count will generally be much greater than the actual count of PS threads since a single thread may process up to 32 pixels.</td>
</tr>
<tr>
<td>31:0</td>
<td></td>
<td><strong>PS Invocation Count LDW</strong>&lt;br&gt;Reflects a count of the total number of pixels (including unlit &quot;helper pixels&quot; within a subspan that need to go through the PS shader to provide 2x2 gradients) that are dispatched to pixel shader invocations while Statistics Enable is set in the Windower in Slice0. See the Windower chapter of the 3D volume for details. This count will generally be much greater than the actual count of PS threads since a single thread may process up to 32 pixels.</td>
</tr>
</tbody>
</table>
PS Invocation Count for Slice1

<table>
<thead>
<tr>
<th>PS_INVOCATION_COUNT_SLICE1 - PS Invocation Count for Slice1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: RenderCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000, 0x00000000</td>
</tr>
<tr>
<td>Access: R/W</td>
</tr>
<tr>
<td>Size (in bits): 64</td>
</tr>
<tr>
<td>Trusted Type: 1</td>
</tr>
<tr>
<td>Address: 022F0h</td>
</tr>
</tbody>
</table>

This register stores the value of the count of pixels that get shaded in Slice0. This register is part of the context save and restore. This register should not be programmed by SW. Note: In GT3 mode there are two slices Slice0 (Upper Slice) and Slice1 (Lower slice).

### Workaround

Workaround: HW reports this count 4X the actual value and therefore SW must divide the count by 4 for correct reporting.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
</table>
| 0..1  | 63:32 | **PS Invocation Count UDW**  
Reflects a count of the total number of pixels (including unlit "helper pixels" within a subspan that need to go through the PS shader to provide 2x2 gradients) that are dispatched to pixel shader invocations while Statistics Enable is set in the Windower in Slice1. See the Windower chapter of the 3D volume for details. This count will generally be much greater than the actual count of PS threads since a single thread may process up to 32 pixels. |
| 31:0  |    | **PS Invocation Count LDW**  
Reflects a count of the total number of pixels (including unlit "helper pixels" within a subspan that need to go through the PS shader to provide 2x2 gradients) that are dispatched to pixel shader invocations while Statistics Enable is set in the Windower in Slice1. See the Windower chapter of the 3D volume for details. This count will generally be much greater than the actual count of PS threads since a single thread may process up to 32 pixels. |
PS Invocation Count for Slice2

<table>
<thead>
<tr>
<th>PS_INVOCATION_COUNT_SLICE2 - PS Invocation Count for Slice2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: RenderCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000, 0x00000000</td>
</tr>
<tr>
<td>Access: R/W</td>
</tr>
<tr>
<td>Size (in bits): 64</td>
</tr>
<tr>
<td>Trusted Type: 1</td>
</tr>
<tr>
<td>Address: 02448h</td>
</tr>
</tbody>
</table>

This register stores the value of the count of pixels that get shaded in Slice2. This register is part of the render context save and restore. This register should not be programmed by SW.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0..1</td>
<td>63:32</td>
<td><strong>PS Invocation Count UDW</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reflects a count of the total number of pixels (including unlit &quot;helper pixels&quot; within a subspan that need to go through the PS shader to provide 2x2 gradients) that are dispatched to pixel shader invocations while Statistics Enable is set in the Windower in Slice2. See the Windower chapter of the 3D volume for details. This count will generally be much greater than the actual count of PS threads since a single thread may process up to 32 pixels.</td>
</tr>
<tr>
<td>31:0</td>
<td></td>
<td><strong>PS Invocation Count LDW</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reflects a count of the total number of pixels (including unlit &quot;helper pixels&quot; within a subspan that need to go through the PS shader to provide 2x2 gradients) that are dispatched to pixel shader invocations while Statistics Enable is set in the Windower in Slice2. See the Windower chapter of the 3D volume for details. This count will generally be much greater than the actual count of PS threads since a single thread may process up to 32 pixels.</td>
</tr>
</tbody>
</table>
## PS Invocation Count for Slice3

<table>
<thead>
<tr>
<th>PS_INVOCATION_COUNT_SLICE3 - PS Invocation Count for Slice3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: RenderCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000, 0x00000000</td>
</tr>
<tr>
<td>Access: R/W</td>
</tr>
<tr>
<td>Size (in bits): 64</td>
</tr>
<tr>
<td>Trusted Type: 1</td>
</tr>
<tr>
<td>Address: 02458h</td>
</tr>
</tbody>
</table>

This register stores the value of the count of pixels that get shaded in Slice3. This register is part of the render context save and restore. This register should not be programmed by SW.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
</table>
| 0..1  | 63:32 | **PS Invocation Count UDW**  
Reflects a count of the total number of pixels (including unlit "helper pixels" within a subspan that need to go through the PS shader to provide 2x2 gradients) that are dispatched to pixel shader invocations while Statistics Enable is set in the Windower in Slice3. See the Windower chapter of the 3D volume for details. This count will generally be much greater than the actual count of PS threads since a single thread may process up to 32 pixels. |
| 31:0  | **PS Invocation Count LDW**  
Reflects a count of the total number of pixels (including unlit "helper pixels" within a subspan that need to go through the PS shader to provide 2x2 gradients) that are dispatched to pixel shader invocations while Statistics Enable is set in the Windower in Slice3. See the Windower chapter of the 3D volume for details. This count will generally be much greater than the actual count of PS threads since a single thread may process up to 32 pixels. |
**PTBR_PAGE_POOL_OOM_EVENT_REGISTER**

<table>
<thead>
<tr>
<th>PTBR_PAGE_POOL_OOM_EVENT_REGISTER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PositionCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 1A5A4h</td>
</tr>
<tr>
<td>Name: Register Template Address</td>
</tr>
<tr>
<td>ShortName: ADDR_SHORT_NAME</td>
</tr>
</tbody>
</table>

Indicates the number of times the POSH pipe has encountered Out of Memory at any given point of time. This register is engine context save/restored.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:17</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 000000000000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
</tbody>
</table>
PTBR Page Pool Size on Out Of Memory

**PTBR_PAGE_POOL_SIZE_ON_OOM_REGISTER**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PositionCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>18594h</td>
</tr>
<tr>
<td>Name:</td>
<td>Register Template Address</td>
</tr>
<tr>
<td>ShortName:</td>
<td>PTBR_PAGE_POOL_SIZE_ON_OOM_REGISTER</td>
</tr>
</tbody>
</table>

Indicates the PTBR_PAGE_POOL_SIZE when POSH pipe has encountered out of memory. This register is engine context save/restored.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:17</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 000000000000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
</tbody>
</table>
### PTE SW Fault Repair High

<table>
<thead>
<tr>
<th>PTESWC_H - PTE SW Fault Repair High</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 04104h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>PTE SW Fault Repair High</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

Fixed PTE entry is written by SW here.
# PTE SW Fault Repair Low

<table>
<thead>
<tr>
<th><strong>PTESWC_L - PTE SW Fault Repair Low</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 04100h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>DWORD</strong></th>
<th><strong>Bit</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>PTE SW Fault Repair Low</strong></td>
</tr>
</tbody>
</table>

- **Default Value:** 00000000h
- **Access:** R/W

Fixed PTE entry is written by SW here.
# Punit to Gunit Message

## P2GMESSAGE - Punit to Gunit Message

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>1300A8h</td>
</tr>
</tbody>
</table>

This register is used for messaging communication between Punit and Gunit.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td>P2GMSGMSK</td>
</tr>
<tr>
<td></td>
<td>Default Value:</td>
<td>0000h</td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>RO</td>
</tr>
</tbody>
</table>

Mask bits for lower 16 bits to avoid a read/modify write
if '0' the corresponding bit in [15:0] is not changed
if '1' the corresponding bit in [15:0] is changed to the value in [15:0]

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td></td>
<td>Punit to Gunit Ack for (EU/SS) resource profile change</td>
</tr>
<tr>
<td></td>
<td>Default Value:</td>
<td>0b</td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>WO</td>
</tr>
</tbody>
</table>

New for CHV, BSW.
Punit sends an ack to proceed with changing the EU/SS requested by CS.
This is self clearing bit. Both this bit and bit 31 should be '1' and byte enables should be set for writing into this bit.
Once written, ack will stay high for one czclk wide and self clears. Reads will always return '0'.
Other bits that are involved in the flow are A11C[28] and A11C[27].Sequencing.
1) CS sends (EU/SS) resource request to GPM.
2) GPM writes to offset 0xDA in Punit config space showing what the intended new configuration will be, when config bit allows.
3) Punit has opportunity to do whatever is needed, but can not wait for CPDack as it will result in a hang.
4) Based on a config unit setting, GPM will wait for an acknowledge from Punit. By default, GPM does not wait for ack due to large latency.
5) GPM continues to change the (EU/SS) resource profile.
6) GPM replies to CS with the new resource profile.
7) Execute the workload or arbitrate CPD.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>14:1</td>
<td></td>
<td>P2G_MSG_RSVD</td>
</tr>
<tr>
<td></td>
<td>Default Value:</td>
<td>0000h</td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>R/W</td>
</tr>
</tbody>
</table>

P2GMSGReserved
<table>
<thead>
<tr>
<th>0</th>
<th>P2GMSG0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Value:</td>
<td>0b</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
</tbody>
</table>

Context restore. This bit is used as part of the hardware context restore process.
Step 1: Punit writes '1' with mask bit 16 set to initiate the HW context restore (for registers driver would normally restore)
Step 2: Gunit goes through context restore.
Step 3: Gunit writes a 1 to Punit register DB[0] indicating context restore has completed.
This bit is cleared automatically when Gunit is powered down. Punit must do the same.
The specific usage of this bit limits it to only be set once, for initial context restore request after an s0ix exit. Any subsequent write is undefined.
### PWRCTXSAVE Message Register for Power Management Unit

**MSG_PWRCTXSAVE_GPM** - PWRCTXSAVE Message Register for Power Management Unit

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>16</td>
</tr>
<tr>
<td>Address:</td>
<td>08044h</td>
</tr>
</tbody>
</table>

Message registers have bit-wise masking applied for writes. The register consists of 16 bits of data in [15:0], and 16 bits of corresponding masks in [31:16]. In order to set bit0, for example, the data would be 0x0001_0001. In order to clear bit0, for example, the data would be 0x0001_0000. Note that mask bit is the data bit offset + 16. Message registers are protected from non-GT writes via the Message Channel.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>15:10</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td><strong>Power Context Save Request</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Power Context Save Request</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1'b0 : Power context save is not being requested &lt;default&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1'b1 : Power context save is being requested</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Unit needs to self-clear this bit upon sampling.</td>
</tr>
<tr>
<td>8:0</td>
<td></td>
<td><strong>QWord Credits for Power Context Save Request</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>QWord Credits for Power Context Save Request</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Minimum Credits = 1 : Unit may send 1 QWord pair (enough for first LRI at least)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Maximum Credits = 511 : Unit may send 511 QWord pairs</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A QWord pair is defined as a 32-bit register address and the corresponding 32-bits of register data. Note that the LRI header and END commands are 64-bits each (32-bit command followed by 32-bit NOOP) and will consume one QWord credit.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Only valid with PWRCTXSAVE_REQ (Bit9).</td>
</tr>
</tbody>
</table>
**PWRMTR_WT1_EEVT1TO4**

<table>
<thead>
<tr>
<th>PWRMTR_WT1_EEVT1TO4 - PWRMTR_WT1_EEVT1TO4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 0A850h</td>
</tr>
</tbody>
</table>

**PWRMTR_WT1_EEVT1to4**

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td>PWRMTR_WT1_EU_TH_EVT4</td>
</tr>
<tr>
<td></td>
<td>Default Value: 00h</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access: R/W Lock</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PWRMTR_WT1_EU_TH_EVT4 This event measured by this field can be overwritten by AACC[3].</td>
<td></td>
</tr>
<tr>
<td>23:16</td>
<td>PWRMTR_WT1_EU_GA_EVT3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default Value: 00h</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access: R/W Lock</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PWRMTR_WT1_EU_GA_EVT3 This event measured by this field can be overwritten by AACC[2].</td>
<td></td>
</tr>
<tr>
<td>15:8</td>
<td>PWRMTR_WT1_EU_GA_EVT2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default Value: 00h</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access: R/W Lock</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PWRMTR_WT1_EU_GA_EVT2 This event measured by this field can be overwritten by AACC[1].</td>
<td></td>
</tr>
<tr>
<td>7:0</td>
<td>PWRMTR_WT1_EU_GA_EVT1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default Value: 00h</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access: R/W Lock</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PWRMTR_WT1_EU_GA_EVT1 This event measured by this field can be overwritten by AACC[0].</td>
<td></td>
</tr>
</tbody>
</table>
### PWRMTR_WT1_EEVT5TO8

**Register Space:** MMIO: 0/2/0  
**Project:** CHV, BSW  
**Source:** PRM  
**Default Value:** 0x00000000  
**Size (in bits):** 32  
**Address:** 0A854h  

#### PWRMTR_WT1_EEVT5to8

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>PWRMTR_WT1_E0_HP_EVT8</strong></td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_E0_HP_EVT8 This event measured by this field can be overwritten by AACC[7].</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>23:16</td>
<td><strong>PWRMTR_WT1_E0_HP_EVT7</strong></td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_E0_HP_EVT7 This event measured by this field can be overwritten by AACC[6].</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>15:8</td>
<td><strong>PWRMTR_WT1_E0_HP_EVT6</strong></td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_E0_HP_EVT6 This event measured by this field can be overwritten by AACC[5].</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>7:0</td>
<td><strong>PWRMTR_WT1_EU_IO_EVT5</strong></td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_EU_IO_EVT5 This event measured by this field can be overwritten by AACC[4].</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### PWRMTR_WT1_EEVT9TO12

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td>PWRMTR_WT1_E0_SP_EVT12</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_E0_SP_EVT12 This event measured by this field can be overwritten by AACC[11].</td>
</tr>
<tr>
<td></td>
<td>23:16</td>
<td>PWRMTR_WT1_E0_SP_EVT11</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_E0_SP_EVT11 This event measured by this field can be overwritten by AACC[10].</td>
</tr>
<tr>
<td></td>
<td>15:8</td>
<td>PWRMTR_WT1_E0_SP_EVT10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_E0_SP_EVT10 This event measured by this field can be overwritten by AACC[9].</td>
</tr>
<tr>
<td></td>
<td>7:0</td>
<td>PWRMTR_WT1_E0_HP_EVT9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_E0_HP_EVT9 This event measured by this field can be overwritten by AACC[8].</td>
</tr>
</tbody>
</table>
## PWRMTR_WT1_EEVT13TO16

<table>
<thead>
<tr>
<th>DWORD</th>
<th>BIT</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td>PWRMTR_WT1_E0_DP_EVT16</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_E0_DP_EVT16 This event measured by this field can be overwritten by AACC[15].</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Programming Notes Must be 0 at all times.</td>
</tr>
<tr>
<td>23:16</td>
<td></td>
<td>PWRMTR_WT1_E0_DP_EVT15</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_E0_DP_EVT15 This event measured by this field can be overwritten by AACC[14].</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Programming Notes Must be 0 at all times.</td>
</tr>
<tr>
<td>15:8</td>
<td></td>
<td>PWRMTR_WT1_E0_DP_EVT14</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_E0_DP_EVT14 This event measured by this field can be overwritten by AACC[13].</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Programming Notes Must be 0 at all times.</td>
</tr>
<tr>
<td>7:0</td>
<td></td>
<td>PWRMTR_WT1_E0_SP_EVT13</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_E0_SP_EVT13 This event measured by this field can be overwritten by AACC[12].</td>
</tr>
</tbody>
</table>
# PWRMTR_WT1_EEVT17TO20

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0A860h</td>
</tr>
</tbody>
</table>

## PWRMTR_WT1_EEVT17to20

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>PWRMTR_WT1_E0INT_EVT20</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>PWRMTR_WT1_E0INT_EVT20</strong> This event measured by this field can be overwritten by AACC[19].</td>
</tr>
<tr>
<td>23:16</td>
<td></td>
<td><strong>PWRMTR_WT1_E0INT_EVT19</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>PWRMTR_WT1_E0INT_EVT19</strong> This event measured by this field can be overwritten by AACC[18].</td>
</tr>
</tbody>
</table>

### Programming Notes

Must be 0 at all times.

| 15:8  |       | **PWRMTR_WT1_E0INT_EVT18**                                                   |
|       |       | Default Value: 00h                                                           |
|       |       | Access: R/W Lock                                                             |
|       |       | **PWRMTR_WT1_E0INT_EVT18** This event measured by this field can be overwritten by AACC[17]. |

| 7:0   |       | **PWRMTR_WT1_E0_DP_EVT17**                                                   |
|       |       | Default Value: 00h                                                           |
|       |       | Access: R/W Lock                                                             |
|       |       | **PWRMTR_WT1_E0_DP_EVT17** This event measured by this field can be overwritten by AACC[16]. |

### Programming Notes

Must be 0 at all times.
## PWRMTR_WT1_EEVT21TO24

### Register Details
- **Register Space:** MMIO: 0/2/0
- **Project:** CHV, BSW
- **Source:** PRM
- **Default Value:** 0x00000000
- **Size (in bits):** 32
- **Address:** 0A864h

### DWord and Bit Details

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td>PWRMTR_WT1_E0_QP_EVT24</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_E0_QP_EVT24 This event measured by this field can be overwritten by AACC[23].</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Programming Notes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Must be 0 at all times.</td>
</tr>
<tr>
<td>23:16</td>
<td></td>
<td>PWRMTR_WT1_E0_QP_EVT23</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_E0_QP_EVT23 This event measured by this field can be overwritten by AACC[22].</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Programming Notes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Must be 0 at all times.</td>
</tr>
<tr>
<td>15:8</td>
<td></td>
<td>PWRMTR_WT1_E0_QP_EVT22</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_E0_QP_EVT22 This event measured by this field can be overwritten by AACC[21].</td>
</tr>
<tr>
<td>7:0</td>
<td></td>
<td>PWRMTR_WT1_E0INT_EVT21</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_E0INT_EVT21 This event measured by this field can be overwritten by AACC[20].</td>
</tr>
</tbody>
</table>
## PWRMTR_WT1_EEVT25TO28

**Register Space:** MMIO: 0/2/0  
**Project:** CHV, BSW  
**Source:** PRM  
**Default Value:** 0x00000000  
**Size (in bits):** 32  
**Address:** 0A868h

### PWRMTR_WT1_EEVT25to28

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td>PWRMTR_WT1_EFPU0_EVT28</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td>23:16</td>
<td>PWRMTR_WT1_EFPU0_EVT27</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td>15:8</td>
<td>PWRMTR_WT1_EFPU0_EVT26</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td>7:0</td>
<td>PWRMTR_WT1_E0_QP_EVT25</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
</tbody>
</table>

### Programming Notes

Must be 0 at all times.
## PWRMTR_WT1_EEVT29TO32

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>31:24</td>
<td>PWRMTR_WT1_E1_HP_EVT32</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_E1_HP_EVT32 This event measured by this field can be overwritten by AACC[31].</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>23:16</td>
<td>PWRMTR_WT1_E1_HP_EVT31</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_E1_HP_EVT31 This event measured by this field can be overwritten by AACC[30].</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>15:8</td>
<td>PWRMTR_WT1_E1_HP_EVT30</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_E1_HP_EVT30 This event measured by this field can be overwritten by AACC[29].</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>7:0</td>
<td>PWRMTR_WT1_EFPU0_EVT29</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_EFPU0_EVT29 This event measured by this field can be overwritten by AACC[28].</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## PWRMTR_WT1_EEVT33TO36

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>31:24</td>
<td><strong>PWRMTR_WT1_E1_SP_EVT36</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_E1_SP_EVT36 This event measured by this field can be overwritten by AAD0[3].</td>
</tr>
<tr>
<td></td>
<td>23:16</td>
<td><strong>PWRMTR_WT1_E1_SP_EVT35</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_E1_SP_EVT35 This event measured by this field can be overwritten by AAD0[2].</td>
</tr>
<tr>
<td></td>
<td>15:8</td>
<td><strong>PWRMTR_WT1_E1_SP_EVT34</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_E1_SP_EVT34 This event measured by this field can be overwritten by AAD0[1].</td>
</tr>
<tr>
<td></td>
<td>7:0</td>
<td><strong>PWRMTR_WT1_E1_HP_EVT33</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_E1_HP_EVT33 This event measured by this field can be overwritten by AAD0[0].</td>
</tr>
</tbody>
</table>
## PWRMTR_WT1_EEVT37TO40

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>PWRMTR_WT1_E1_DP_EVT40</strong></td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_E1_DP_EVT40 This event measured by this field can be overwritten by AAD0[7].</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23:16</td>
<td></td>
<td><strong>PWRMTR_WT1_E1_DP_EVT39</strong></td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_E1_DP_EVT39 This event measured by this field can be overwritten by AAD0[6].</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15:8</td>
<td></td>
<td><strong>PWRMTR_WT1_E1_DP_EVT38</strong></td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_E1_DP_EVT38 This event measured by this field can be overwritten by AAD0[5].</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7:0</td>
<td></td>
<td><strong>PWRMTR_WT1_E1_SP_EVT37</strong></td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_E1_SP_EVT37 This event measured by this field can be overwritten by AAD0[4].</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# PWRMTR_WT1_EEVT41TO44

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>PWRMTR_WT1_E1INT_EVT44</strong></td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_E1INT_EVT44 This event measured by this field can be overwritten by AAD0[11].</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>23:16</td>
<td><strong>PWRMTR_WT1_E1INT_EVT43</strong></td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_E1INT_EVT43 This event measured by this field can be overwritten by AAD0[10].</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>15:8</td>
<td><strong>PWRMTR_WT1_E1INT_EVT42</strong></td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_E1INT_EVT42 This event measured by this field can be overwritten by AAD0[9].</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>7:0</td>
<td><strong>PWRMTR_WT1_E1_DP_EVT41</strong></td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_E1_DP_EVT41 This event measured by this field can be overwritten by AAD0[8].</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Programming Notes**

Must be 0 at all times.
### PWRMTR_WT1_EEVT45TO48

<table>
<thead>
<tr>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>PWRMTR_WT1_E1_QP_EVT48</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td>PWRMTR_WT1_E1_QP_EVT47</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td>PWRMTR_WT1_E1_QP_EVT46</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td>PWRMTR_WT1_E1INT_EVT45</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
</tbody>
</table>

#### Programming Notes
- Must be 0 at all times.
- PWRMTR_WT1_E1_QP_EVT48 This event measured by this field can be overwritten by AAD0[15].
- PWRMTR_WT1_E1_QP_EVT47 This event measured by this field can be overwritten by AAD0[14].
- PWRMTR_WT1_E1_QP_EVT46 This event measured by this field can be overwritten by AAD0[13].
- PWRMTR_WT1_E1INT_EVT45 This event measured by this field can be overwritten by AAD0[12].
# PWRMTR_WT1_EEVT49TO52

## Register Details

- **Register Space:** MMIO: 0/2/0
- **Project:** CHV, BSW
- **Source:** PRM
- **Default Value:** 0x00000000
- **Size (in bits):** 32
- **Address:** 0A880h

## DWord Bit Description

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Default Value</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td>PWRMTR_WT1_EFPU1_EVT52</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td>23:16</td>
<td>PWRMTR_WT1_EFPU1_EVT51</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td>15:8</td>
<td>PWRMTR_WT1_EFPU1_EVT50</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td>7:0</td>
<td>PWRMTR_WT1_E1_QP_EVT49</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
</tbody>
</table>
### PWRMTR_WT1_EEVT53TO56

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>PWRMTR_WT1_EEMEM_EVT56</strong>&lt;br&gt;Default Value: 00h&lt;br&gt;Access: R/W Lock&lt;br&gt;PWRMTR_WT1_EEMEM_EVT56 This event measured by this field can be overwritten by AAD0[23].</td>
</tr>
<tr>
<td></td>
<td>23:16</td>
<td><strong>PWRMTR_WT1_EEMEM_EVT55</strong>&lt;br&gt;Default Value: 00h&lt;br&gt;Access: R/W Lock&lt;br&gt;PWRMTR_WT1_EEMEM_EVT55 This event measured by this field can be overwritten by AAD0[22].</td>
</tr>
<tr>
<td></td>
<td>15:8</td>
<td><strong>PWRMTR_WT1_EEMEM_EVT54</strong>&lt;br&gt;Default Value: 00h&lt;br&gt;Access: R/W Lock&lt;br&gt;PWRMTR_WT1_EEMEM_EVT54 This event measured by this field can be overwritten by AAD0[21].</td>
</tr>
<tr>
<td></td>
<td>7:0</td>
<td><strong>PWRMTR_WT1_EFPU1_EVT53</strong>&lt;br&gt;Default Value: 00h&lt;br&gt;Access: R/W Lock&lt;br&gt;PWRMTR_WT1_EFPU1_EVT53 This event measured by this field can be overwritten by AAD0[20].</td>
</tr>
</tbody>
</table>
## PWRMTR_WT1_EEVT57TO58

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td>Reserved</td>
<td>0000h</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
<td>0000h</td>
<td>RO</td>
</tr>
<tr>
<td>15:8</td>
<td></td>
<td><strong>PWRMTR_WT1_EBUSS_EVT58</strong></td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_EBUSS_EVT58 This event measured by this field can be overwritten by AAD0[25].</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7:0</td>
<td></td>
<td><strong>PWRMTR_WT1_EEMEM_EVT57</strong></td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_EEMEM_EVT57 This event measured by this field can be overwritten by AAD0[24].</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## PWRMTR_WT1_MEVT1TO4

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>PWRMTR_WT1_MMDOI_EVT4</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_MMDOI_EVT4 This event measured by this field can be overwritten by AAD4[3].</td>
</tr>
<tr>
<td></td>
<td>23:16</td>
<td><strong>PWRMTR_WT1_MMSOI_EVT3</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_MMSOI_EVT3 This event measured by this field can be overwritten by AAD4[2].</td>
</tr>
<tr>
<td></td>
<td>15:8</td>
<td><strong>PWRMTR_WT1_MMSOI_EVT2</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_MMSOI_EVT2 This event measured by this field can be overwritten by AAD4[1].</td>
</tr>
<tr>
<td></td>
<td>7:0</td>
<td><strong>PWRMTR_WT1_MMSOI_EVT1</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_MMSOI_EVT1 This event measured by this field can be overwritten by AAD4[0].</td>
</tr>
</tbody>
</table>
### PWRMTR_WT1_MEVT5TO8

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>PWRMTR_WT1_MMDOI_EVT8</strong>&lt;br&gt;Default Value: 00h&lt;br&gt;Access: R/W Lock&lt;br&gt;PWRMTR_WT1_MMDOI_EVT8 This event measured by this field can be overwritten by AAD4[7].</td>
</tr>
<tr>
<td>23:16</td>
<td>PWRMTR_WT1_MMDOI_EVT7&lt;br&gt;Default Value: 00h&lt;br&gt;Access: R/W Lock&lt;br&gt;PWRMTR_WT1_MMDOI_EVT7 This event measured by this field can be overwritten by AAD4[6].</td>
<td></td>
</tr>
<tr>
<td>15:8</td>
<td>PWRMTR_WT1_MMDOI_EVT6&lt;br&gt;Default Value: 00h&lt;br&gt;Access: R/W Lock&lt;br&gt;PWRMTR_WT1_MMDOI_EVT6 This event measured by this field can be overwritten by AAD4[5].</td>
<td></td>
</tr>
<tr>
<td>7:0</td>
<td>PWRMTR_WT1_MMDOI_EVT5&lt;br&gt;Default Value: 00h&lt;br&gt;Access: R/W Lock&lt;br&gt;PWRMTR_WT1_MMDOI_EVT5 This event measured by this field can be overwritten by AAD4[4].</td>
<td></td>
</tr>
</tbody>
</table>
## PWRMTR_WT1_MEVT9T012

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong></td>
<td>MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong></td>
<td>CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong></td>
<td>PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong></td>
<td>0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong></td>
<td>32</td>
</tr>
<tr>
<td><strong>Address:</strong></td>
<td>0A898h</td>
</tr>
</tbody>
</table>

### PWRMTR_WT1_MEVT9to12

<table>
<thead>
<tr>
<th>DW</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>PWRMTR_WT1_MMDCN_EVT12</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>PWRMTR_WT1_MMDCN_EVT12</strong> This event measured by this field can be overwritten by AAD4[11].</td>
</tr>
<tr>
<td></td>
<td>23:16</td>
<td><strong>PWRMTR_WT1_MMDCN_EVT11</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>PWRMTR_WT1_MMDCN_EVT11</strong> This event measured by this field can be overwritten by AAD4[10].</td>
</tr>
<tr>
<td></td>
<td>15:8</td>
<td><strong>PWRMTR_WT1_MMDCN_EVT10</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>PWRMTR_WT1_MMDCN_EVT10</strong> This event measured by this field can be overwritten by AAD4[9].</td>
</tr>
<tr>
<td></td>
<td>7:0</td>
<td><strong>PWRMTR_WT1_MMDCN_EVT9</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>PWRMTR_WT1_MMDCN_EVT9</strong> This event measured by this field can be overwritten by AAD4[8].</td>
</tr>
</tbody>
</table>
### PWRMTR_WT1_MEVT13TO16 - PWRMTR_WT1_MEVT13TO16

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
</table>
| 0     | 31:24  | **PWRMTR_WT1_MMDCN_EVT16**  
Default Value: 00h  
Access: R/W Lock  
PWRMTR_WT1_MMDCN_EVT16 This event measured by this field can be overwritten by AAD4[15]. |
| 23:16 |        | **PWRMTR_WT1_MMDCN_EVT15**  
Default Value: 00h  
Access: R/W Lock  
PWRMTR_WT1_MMDCN_EVT15 This event measured by this field can be overwritten by AAD4[14]. |
| 15:8  |        | **PWRMTR_WT1_MMDCN_EVT14**  
Default Value: 00h  
Access: R/W Lock  
PWRMTR_WT1_MMDCN_EVT14 This event measured by this field can be overwritten by AAD4[13]. |
| 7:0   |        | **PWRMTR_WT1_MMDCN_EVT13**  
Default Value: 00h  
Access: R/W Lock  
PWRMTR_WT1_MMDCN_EVT13 This event measured by this field can be overwritten by AAD4[12]. |
## PWRMTR_WT1_MEVT17TO20

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>PWRMTR_WT1_MMDMH_EVT20</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_MMDMH_EVT20 This event measured by this field can be overwritten by AAD4[19].</td>
</tr>
<tr>
<td>23:16</td>
<td></td>
<td><strong>PWRMTR_WT1_MMDCN_EVT19</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_MMDCN_EVT19 This event measured by this field can be overwritten by AAD4[18].</td>
</tr>
<tr>
<td>15:8</td>
<td></td>
<td><strong>PWRMTR_WT1_MMDCN_EVT18</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_MMDCN_EVT18 This event measured by this field can be overwritten by AAD4[17].</td>
</tr>
<tr>
<td>7:0</td>
<td></td>
<td><strong>PWRMTR_WT1_MMDCN_EVT17</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_MMDCN_EVT17 This event measured by this field can be overwritten by AAD4[16].</td>
</tr>
</tbody>
</table>
# PWRMTR_WT1_MEVT21TO24

## Register Space:
- MMIO: 0/2/0

## Project:
- CHV, BSW

## Source:
- PRM

## Default Value:
- 0x00000000

## Size (in bits):
- 32

## Address:
- 0A8A4h

## Description:

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>PWRMTR_WT1_MMDOH_EVT24</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_MMDOH_EVT24 This event measured by this field can be overwritten by AAD4[23].</td>
</tr>
<tr>
<td>23:16</td>
<td></td>
<td><strong>PWRMTR_WT1_MMDMH_EVT23</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_MMDMH_EVT23 This event measured by this field can be overwritten by AAD4[22].</td>
</tr>
<tr>
<td>15:8</td>
<td></td>
<td><strong>PWRMTR_WT1_MMDMH_EVT22</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_MMDMH_EVT22 This event measured by this field can be overwritten by AAD4[21].</td>
</tr>
<tr>
<td>7:0</td>
<td></td>
<td><strong>PWRMTR_WT1_MMDMH_EVT21</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_MMDMH_EVT21 This event measured by this field can be overwritten by AAD4[20].</td>
</tr>
</tbody>
</table>
## PWRMTR_WT1_MEVT25TO28

<table>
<thead>
<tr>
<th>DW0</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>31:24</td>
<td>PWRMTR_WT1_MMDCN_EVT28</td>
<td>Default Value: 00h</td>
<td>R/W Lock</td>
<td></td>
</tr>
<tr>
<td>23:16</td>
<td>PWRMTR_WT1_MMDCN_EVT27</td>
<td>Default Value: 00h</td>
<td>R/W Lock</td>
<td></td>
</tr>
<tr>
<td>15:8</td>
<td>PWRMTR_WT1_MMDCN_EVT26</td>
<td>Default Value: 00h</td>
<td>R/W Lock</td>
<td></td>
</tr>
<tr>
<td>7:0</td>
<td>PWRMTR_WT1_MMDOH_EVT25</td>
<td>Default Value: 00h</td>
<td>R/W Lock</td>
<td></td>
</tr>
</tbody>
</table>
# PWRMTR_WT1_MEVT29TO32

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td>PWRMTR_WT1_MHECN_EVT32</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_MHECN_EVT32</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>This event measured by this field can be overwritten by AAD4[31].</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23:16</td>
<td></td>
<td>PWRMTR_WT1_MHECN_EVT31</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_MHECN_EVT31</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>This event measured by this field can be overwritten by AAD4[30].</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15:8</td>
<td></td>
<td>PWRMTR_WT1_MHECN_EVT30</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_MHECN_EVT30</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>This event measured by this field can be overwritten by AAD4[29].</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7:0</td>
<td></td>
<td>PWRMTR_WT1_MMDCN_EVT29</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_MMDCN_EVT29</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>This event measured by this field can be overwritten by AAD4[28].</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## PWRMTR_WT1_MEVT33TO36

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
</table>
| 0     | 31:24 | **PWRMTR_WT1_MHECN_EVT36**  
  Default Value: 00h  
  Access: R/W Lock  
PWRMTR_WT1_MHECN_EVT36 This event measured by this field can be overwritten by AAD8[3]. |
|       | 23:16 | **PWRMTR_WT1_MHECN_EVT35**  
  Default Value: 00h  
  Access: R/W Lock  
PWRMTR_WT1_MHECN_EVT35 This event measured by this field can be overwritten by AAD8[2]. |
|       | 15:8  | **PWRMTR_WT1_MHECN_EVT34**  
  Default Value: 00h  
  Access: R/W Lock  
PWRMTR_WT1_MHECN_EVT34 This event measured by this field can be overwritten by AAD8[1]. |
|       | 7:0   | **PWRMTR_WT1_MHECN_EVT33**  
  Default Value: 00h  
  Access: R/W Lock  
PWRMTR_WT1_MHECN_EVT33 This event measured by this field can be overwritten by AAD8[0]. |
## PWRMTR_WT1_REVT1TO4

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0A800h</td>
</tr>
</tbody>
</table>

### PWRMTR_WT1_REVT1to4

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>PWRMTR_WT1_RFFOI_EVT4</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_RFFOI_EVT4. This event measured by this field can be overwritten by AAC0[3].</td>
</tr>
<tr>
<td>23:16</td>
<td><strong>PWRMTR_WT1_RFFOI_EVT3</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_RFFOI_EVT3. This event measured by this field can be overwritten by AAC0[2].</td>
</tr>
<tr>
<td>15:8</td>
<td><strong>PWRMTR_WT1_RFFOI_EVT2</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_RFFOI_EVT2. This event measured by this field can be overwritten by AAC0[1].</td>
</tr>
<tr>
<td>7:0</td>
<td><strong>PWRMTR_WT1_RFFOI_EVT1</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_RFFOI_EVT1. This event measured by this field can be overwritten by AAC0[0].</td>
</tr>
</tbody>
</table>
## PWRMTR_WT1_REVT5TO8

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
</table>
| 0     | 31:24 | **PWRMTR_WT1_RFFOI_EVT8**  
Default Value: 00h  
Access: R/W Lock  
PWRMTR_WT1_RFFOI_EVT8. This event measured by this field can be overwritten by AAC0[7]. |
| 23:16 | **PWRMTR_WT1_RFFOI_EVT7**  
Default Value: 00h  
Access: R/W Lock  
PWRMTR_WT1_RFFOI_EVT7. This event measured by this field can be overwritten by AAC0[6]. |
| 15:8  | **PWRMTR_WT1_RFFOI_EVT6**  
Default Value: 00h  
Access: R/W Lock  
PWRMTR_WT1_RFFOI_EVT6. This event measured by this field can be overwritten by AAC0[5]. |
| 7:0   | **PWRMTR_WT1_RFFOI_EVT5**  
Default Value: 00h  
Access: R/W Lock  
PWRMTR_WT1_RFFOI_EVT5. This event measured by this field can be overwritten by AAC0[4]. |
### PWRMTR_WT1_REVT9TO12

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td>PVRMTR_WT1_RFFOI_EVT12</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_RFFOI_EVT12. This event measured by this field can be overwritten by AAC0[11].</td>
</tr>
<tr>
<td></td>
<td>23:16</td>
<td>PWRMTR_WT1_RFFOI_EVT11</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_RFFOI_EVT11. This event measured by this field can be overwritten by AAC0[10].</td>
</tr>
<tr>
<td></td>
<td>15:8</td>
<td>PWRMTR_WT1_RFFOI_EVT10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_RFFOI_EVT10. This event measured by this field can be overwritten by AAC0[9].</td>
</tr>
<tr>
<td></td>
<td>7:0</td>
<td>PWRMTR_WT1_RFFOI_EVT9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_RFFOI_EVT9. This event measured by this field can be overwritten by AAC0[8].</td>
</tr>
</tbody>
</table>
### PWRMTR_WT1_REVT13TO16 - PWRMTR_WT1_REVT13TO16

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0A80Ch</td>
</tr>
</tbody>
</table>

#### PWRMTR_WT1_REVT13to16

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
<th>Programming Notes</th>
<th>Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>PWRMTR_WT1_RFFCN_EVT16</strong></td>
<td>00h</td>
<td>R/W Lock</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>PWRMTR_WT1_RFFCN_EVT16.</strong> This event measured by this field can be overwritten by AAC0[15].</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Must be 0 at all times.</strong></td>
<td></td>
<td></td>
<td></td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>23:16</td>
<td></td>
<td><strong>PWRMTR_WT1_RFFCN_EVT15</strong></td>
<td>00h</td>
<td>R/W Lock</td>
<td><strong>PWRMTR_WT1_RFFCN_EVT15.</strong> This event measured by this field can be overwritten by AAC0[14].</td>
<td></td>
</tr>
<tr>
<td>15:8</td>
<td></td>
<td><strong>PWRMTR_WT1_RFFOI_EVT14</strong></td>
<td>00h</td>
<td>R/W Lock</td>
<td><strong>PWRMTR_WT1_RFFOI_EVT14.</strong> This event measured by this field can be overwritten by AAC0[13].</td>
<td></td>
</tr>
<tr>
<td>7:0</td>
<td></td>
<td><strong>PWRMTR_WT1_RFFOI_EVT13</strong></td>
<td>00h</td>
<td>R/W Lock</td>
<td><strong>PWRMTR_WT1_RFFOI_EVT13.</strong> This event measured by this field can be overwritten by AAC0[12].</td>
<td></td>
</tr>
</tbody>
</table>
## PWRMTR_WT1_REVT17TO20

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>PWRMTR_WT1_RFFCN_EVT20</strong>&lt;br&gt;Default Value: 00h&lt;br&gt;Access: R/W Lock&lt;br&gt;PWRMTR_WT1_RFFCN_EVT20. This event measured by this field can be overwritten by AAC0[19].</td>
</tr>
<tr>
<td>23:16</td>
<td></td>
<td><strong>PWRMTR_WT1_RFFCN_EVT19</strong>&lt;br&gt;Default Value: 00h&lt;br&gt;Access: R/W Lock&lt;br&gt;PWRMTR_WT1_RFFCN_EVT19. This event measured by this field can be overwritten by AAC0[18].</td>
</tr>
<tr>
<td>15:8</td>
<td></td>
<td><strong>PWRMTR_WT1_RFFCN_EVT18</strong>&lt;br&gt;Default Value: 00h&lt;br&gt;Access: R/W Lock&lt;br&gt;PWRMTR_WT1_RFFCN_EVT18. This event measured by this field can be overwritten by AAC0[17].</td>
</tr>
<tr>
<td>7:0</td>
<td></td>
<td><strong>PWRMTR_WT1_RFFCN_EVT17</strong>&lt;br&gt;Default Value: 00h&lt;br&gt;Access: R/W Lock&lt;br&gt;PWRMTR_WT1_RFFCN_EVT17. This event measured by this field can be overwritten by AAC0[16].</td>
</tr>
</tbody>
</table>

### Programming Notes

Must be 0 at all times.

<table>
<thead>
<tr>
<th>Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHV, BSW</td>
</tr>
</tbody>
</table>
# PWRMTR_WT1_REVT21TO24

<table>
<thead>
<tr>
<th><strong>Register Space:</strong></th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Project:</strong></td>
<td>CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong></td>
<td>PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong></td>
<td>0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong></td>
<td>32</td>
</tr>
<tr>
<td><strong>Address:</strong></td>
<td>0A814h</td>
</tr>
</tbody>
</table>

## PWRMTR_WT1_REVT21to24

<table>
<thead>
<tr>
<th><strong>DWord</strong></th>
<th><strong>Bit</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td>PWRMTR_WT1_RSCOI_EVT24</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_RSCOI_EVT24 This event measured by this field can be overwritten by AAC0[23].</td>
</tr>
<tr>
<td></td>
<td>23:16</td>
<td>PWRMTR_WT1_RSCOI_EVT23</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_RSCOI_EVT23 This event measured by this field can be overwritten by AAC0[22].</td>
</tr>
<tr>
<td></td>
<td>15:8</td>
<td>PWRMTR_WT1_RSCOI_EVT22</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_RSCOI_EVT22 This event measured by this field can be overwritten by AAC0[21].</td>
</tr>
<tr>
<td></td>
<td>7:0</td>
<td>PWRMTR_WT1_RSCOI_EVT21</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_RSCOI_EVT21 This event measured by this field can be overwritten by AAC0[20].</td>
</tr>
</tbody>
</table>
## PWRMTR_WT1_REVT25TO28

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td>PWRMTR_WT1_RSCOI_EVT28</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_RSCOI_EVT28 This event measured by this field can be overwritten by AAC0[27].</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>23:16</td>
<td>PWRMTR_WT1_RSCOI_EVT27</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_RSCOI_EVT27 This event measured by this field can be overwritten by AAC0[26].</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>15:8</td>
<td>PWRMTR_WT1_RSCOI_EVT26</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_RSCOI_EVT26 This event measured by this field can be overwritten by AAC0[25].</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>7:0</td>
<td>PWRMTR_WT1_RSCOI_EVT25</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_RSCOI_EVT25 This event measured by this field can be overwritten by AAC0[24].</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## PWRMTR_WT1_REVT29TO32

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>PWRMTR_WT1_RSCOI_EVT32</strong></td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This event measured by this field can be overwritten by AAC[31].</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23:16</td>
<td>PWRMTR_WT1_RSCOI_EVT31</td>
<td>00h</td>
<td>R/W Lock</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>This event measured by this field can be overwritten by AAC[30].</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15:8</td>
<td>PWRMTR_WT1_RSCOI_EVT30</td>
<td>00h</td>
<td>R/W Lock</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>This event measured by this field can be overwritten by AAC[29].</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7:0</td>
<td>PWRMTR_WT1_RSCOI_EVT29</td>
<td>00h</td>
<td>R/W Lock</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>This event measured by this field can be overwritten by AAC[28].</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## PWRMTR_WT1_REVT33TO36

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
</tbody>
</table>

**Address**: 0A820h

**PWRMTR_WT1_REVT33to36**

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>PWRMTR_WT1_RL3MI_EVT36</strong></td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_RL3MI_EVT36 This event measured by this field can be overwritten by AAC4[3].</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23:16</td>
<td></td>
<td><strong>PWRMTR_WT1_RL3CN_EVT35</strong></td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_RL3CN_EVT35 This event measured by this field can be overwritten by AAC4[2].</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15:8</td>
<td></td>
<td><strong>PWRMTR_WT1_RSCCN_EVT34</strong></td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_RSCCN_EVT34 This event measured by this field can be overwritten by AAC4[1].</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7:0</td>
<td></td>
<td><strong>PWRMTR_WT1_RSCOI_EVT33</strong></td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_RSCOI_EVT33 This event measured by this field can be overwritten by AAC4[0].</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## PWRMTR_WT1_REVT37TO40

### Register Summary
- **Register Space:** MMIO: 0/2/0
- **Project:** CHV, BSW
- **Source:** PRM
- **Default Value:** 0x00000000
- **Size (in bits):** 32
- **Address:** 0A824h

### Description
- **PWRMTR_WT1_RTAOI_EVT40**
  - **Default Value:** 00h
  - **Access:** R/W Lock
  - This event measured by this field can be overwritten by AAC4[7].

- **PWRMTR_WT1_RL3MI_EVT39**
  - **Default Value:** 00h
  - **Access:** R/W Lock
  - This event measured by this field can be overwritten by AAC4[6].

- **PWRMTR_WT1_RL3MI_EVT38**
  - **Default Value:** 00h
  - **Access:** R/W Lock
  - This event measured by this field can be overwritten by AAC4[5].

- **PWRMTR_WT1_RL3MI_EVT37**
  - **Default Value:** 00h
  - **Access:** R/W Lock
  - This event measured by this field can be overwritten by AAC4[4].
## PWRMTR_WT1_REVT41TO44

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0A828h</td>
</tr>
</tbody>
</table>

### DWord 0

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>31:24</td>
<td>PWRMTR_WT1_RTAOI_EVT44</td>
</tr>
<tr>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td>PWRMTR_WT1_RTAOI_EVT44 This event measured by this field can be overwritten by AAC4[11].</td>
</tr>
</tbody>
</table>

### DWord 23:16

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>23:16</td>
<td>PWRMTR_WT1_RTAOI_EVT43</td>
</tr>
<tr>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td>PWRMTR_WT1_RTAOI_EVT43 This event measured by this field can be overwritten by AAC4[10].</td>
</tr>
</tbody>
</table>

### DWord 15:8

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>15:8</td>
<td>PWRMTR_WT1_RTAOI_EVT42</td>
</tr>
<tr>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td>PWRMTR_WT1_RTAOI_EVT42 This event measured by this field can be overwritten by AAC4[9].</td>
</tr>
</tbody>
</table>

### DWord 7:0

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7:0</td>
<td>PWRMTR_WT1_RTAOI_EVT41</td>
</tr>
<tr>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td>PWRMTR_WT1_RTAOI_EVT41 This event measured by this field can be overwritten by AAC4[8].</td>
</tr>
</tbody>
</table>
## PWRMTR_WT1_REV45TO48

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>PWRMTR_WT1_RTAOI_EVT48</strong>&lt;br&gt;Default Value: 00h&lt;br&gt;Access: R/W Lock&lt;br&gt;PWRMTR_WT1_RTAOI_EVT48 This event measured by this field can be overwritten by AAC4[15].</td>
</tr>
<tr>
<td>0</td>
<td>23:16</td>
<td><strong>PWRMTR_WT1_RTAOI_EVT47</strong>&lt;br&gt;Default Value: 00h&lt;br&gt;Access: R/W Lock&lt;br&gt;PWRMTR_WT1_RTAOI_EVT47 This event measured by this field can be overwritten by AAC4[14].</td>
</tr>
<tr>
<td>0</td>
<td>15:8</td>
<td><strong>PWRMTR_WT1_RTAOI_EVT46</strong>&lt;br&gt;Default Value: 00h&lt;br&gt;Access: R/W Lock&lt;br&gt;PWRMTR_WT1_RTAOI_EVT46 This event measured by this field can be overwritten by AAC4[13].</td>
</tr>
<tr>
<td>0</td>
<td>7:0</td>
<td><strong>PWRMTR_WT1_RTAOI_EVT45</strong>&lt;br&gt;Default Value: 00h&lt;br&gt;Access: R/W Lock&lt;br&gt;PWRMTR_WT1_RTAOI_EVT45 This event measured by this field can be overwritten by AAC4[12].</td>
</tr>
</tbody>
</table>
# PWRMTR_WT1_REVT49TO52

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
</table>
| 0     | 31:24 | **PWRMTR_WT1_RTAOH_EVT52**  
Default Value: 00h  
Access: R/W Lock  
PWRMTR_WT1_RTAOH_EVT52 This event measured by this field can be overwritten by AAC4[19]. |
| 23:16 | **PWRMTR_WT1_RTAOH_EVT51**  
Default Value: 00h  
Access: R/W Lock  
PWRMTR_WT1_RTAOH_EVT51 This event measured by this field can be overwritten by AAC4[18]. |
| 15:8  | **PWRMTR_WT1_RTAOI_EVT50**  
Default Value: 00h  
Access: R/W Lock  
PWRMTR_WT1_RTAOI_EVT50 This event measured by this field can be overwritten by AAC4[17]. |
| 7:0   | **PWRMTR_WT1_RTAOI_EVT49**  
Default Value: 00h  
Access: R/W Lock  
PWRMTR_WT1_RTAOI_EVT49 This event measured by this field can be overwritten by AAC4[16]. |
## PWRMTR_WT1_REV53TO56

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0A834h</td>
</tr>
</tbody>
</table>

### PWRMTR_WT1_REV53to56

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>PWRMTR_WT1_RTSOL_EVT56</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_RTS OI_EVT56 This event measured by this field can be overwritten by AAC4[23].</td>
</tr>
<tr>
<td>23:16</td>
<td></td>
<td><strong>PWRMTR_WT1_RTACN_EVT55</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_RTACN_EVT55 This event measured by this field can be overwritten by AAC4[22].</td>
</tr>
<tr>
<td>15:8</td>
<td></td>
<td><strong>PWRMTR_WT1_RTACN_EVT54</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_RTACN_EVT54 This event measured by this field can be overwritten by AAC4[21].</td>
</tr>
<tr>
<td>7:0</td>
<td></td>
<td><strong>PWRMTR_WT1_RTACN_EVT53</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_RTACN_EVT53 This event measured by this field can be overwritten by AAC4[20].</td>
</tr>
</tbody>
</table>
## PWRMTR_WT1_REV57TO60

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
<th>Programming Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td>PWRMTR_WT1_RTSOI_EVT60</td>
<td>00h</td>
<td>R/W Lock</td>
<td>This event measured by this field can be overwritten by AAC4[27].</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23:16</td>
<td>PWRMTR_WT1_RTSOI_EVT59</td>
<td>00h</td>
<td>R/W Lock</td>
<td>This event measured by this field can be overwritten by AAC4[26].</td>
<td></td>
</tr>
<tr>
<td>15:8</td>
<td>PWRMTR_WT1_RTSOI_EVT58</td>
<td>00h</td>
<td>R/W Lock</td>
<td>This event measured by this field can be overwritten by AAC4[25].</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7:0</td>
<td>PWRMTR_WT1_RTSOI_EVT57</td>
<td>00h</td>
<td>R/W Lock</td>
<td>This event measured by this field can be overwritten by AAC4[24].</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Must be set to zero at all times.</td>
</tr>
</tbody>
</table>

**Register Space:** MMIO: 0/2/0  
**Project:** CHV, BSW  
**Source:** PRM  
**Default Value:** 0x00000000  
**Size (in bits):** 32  
**Address:** 0A838h
### PWRMTR_WT1_REV61TO64

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td>PWRMTR_WT1_RTOOI_EVT64</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_RTOOI_EVT64 This event measured by this field can be overwritten by AAC4[31].</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>23:16</td>
<td>PWRMTR_WT1_RTOOI_EVT63</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_RTOOI_EVT63 This event measured by this field can be overwritten by AAC4[30].</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>15:8</td>
<td>PWRMTR_WT1_RTOOI_EVT62</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_RTOOI_EVT62 This event measured by this field can be overwritten by AAC4[29].</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>7:0</td>
<td>PWRMTR_WT1_RTOOI_EVT61</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_RTOOI_EVT61 This event measured by this field can be overwritten by AAC4[28].</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### PWRMTR_WT1_REVT65TO68

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0A840h</td>
</tr>
</tbody>
</table>

#### PWRMTR_WT1_REVT65TO68 - PWRMTR_WT1_REVT65TO68

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>PWRMTR_WT1_RGTCN_EVT68</strong></td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_RGTCN_EVT68 This event measured by this field can be overwritten by AAC8[3].</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23:16</td>
<td></td>
<td><strong>PWRMTR_WT1_RGTCN_EVT67</strong></td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_RGTCN_EVT67 This event measured by this field can be overwritten by AAC8[2].</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15:8</td>
<td></td>
<td><strong>PWRMTR_WT1_RGTCN_EVT66</strong></td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_RGTCN_EVT66 This event measured by this field can be overwritten by AAC8[1].</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7:0</td>
<td></td>
<td><strong>PWRMTR_WT1_RGTOI_EVT65</strong></td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_RGTOI_EVT65 This event measured by this field can be overwritten by AAC8[0].</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# PWRMTR_WT1_REVT69TO70

## Register Information
- **Register Space:** MMIO: 0/2/0
- **Project:** CHV, BSW
- **Source:** PRM
- **Default Value:** 0x00000000
- **Size (in bits):** 32
- **Address:** 0A844h

### DWord & Bit Description

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td>15:8</td>
<td><strong>PWRMTR_WT1_RGTCN_EVT70</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_RGTCN_EVT70 This event measured by this field can be overwritten by AAC8[5].</td>
</tr>
<tr>
<td></td>
<td>7:0</td>
<td><strong>PWRMTR_WT1_RGTCN_EVT69</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT1_RGTCN_EVT69 This event measured by this field can be overwritten by AAC8[4].</td>
</tr>
</tbody>
</table>
## PWRMTR_WT2_EEVT1TO4

**Register Space:** MMIO: 0/2/0  
**Project:** CHV, BSW  
**Source:** PRM  
**Default Value:** 0x00000000  
**Size (in bits):** 32

**Address:** 0A950h

### PWRMTR_WT2_EEVT1to4

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>PWRMTR_WT2_EU_TH_EVT4</strong></td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT2_EU_TH_EVT4 This event measured by this field can be overwritten by AACC[3].</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23:16</td>
<td></td>
<td><strong>PWRMTR_WT2_EU_GA_EVT3</strong></td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT2_EU_GA_EVT3 This event measured by this field can be overwritten by AACC[2].</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15:8</td>
<td></td>
<td><strong>PWRMTR_WT2_EU_GA_EVT2</strong></td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT2_EU_GA_EVT2 This event measured by this field can be overwritten by AACC[1].</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7:0</td>
<td></td>
<td><strong>PWRMTR_WT2_EU_GA_EVT1</strong></td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT2_EU_GA_EVT1 This event measured by this field can be overwritten by AACC[0].</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## PWRMTR_WT2_EEVT5TO8

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td>PWRMTR_WT2_E0_HP_EVT8</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT2_E0_HP_EVT8 This event measured by this field can be overwritten by AACC[7].</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>23:16</td>
<td>PWRMTR_WT2_E0_HP_EVT7</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT2_E0_HP_EVT7 This event measured by this field can be overwritten by AACC[6].</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>15:8</td>
<td>PWRMTR_WT2_E0_HP_EVT6</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT2_E0_HP_EVT6 This event measured by this field can be overwritten by AACC[5].</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>7:0</td>
<td>PWRMTR_WT2_EU_IO_EVT5</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT2_EU_IO_EVT5 This event measured by this field can be overwritten by AACC[4].</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### PWRMTR_WT2_EEVT9TO12

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
</table>
| 0     | 31:24 | **PWRMTR_WT2_E0_SP_EVT12**  
|       |       | Default Value: 00h  
|       |       | Access: R/W Lock  
|       |       | PWRMTR_WT2_E0_SP_EVT12 This event measured by this field can be overwritten by AACC[11]. |
|       | 23:16 | **PWRMTR_WT2_E0_SP_EVT11**  
|       |       | Default Value: 00h  
|       |       | Access: R/W Lock  
|       |       | PWRMTR_WT2_E0_SP_EVT11 This event measured by this field can be overwritten by AACC[10]. |
|       | 15:8  | **PWRMTR_WT2_E0_SP_EVT10**  
|       |       | Default Value: 00h  
|       |       | Access: R/W Lock  
|       |       | PWRMTR_WT2_E0_SP_EVT10 This event measured by this field can be overwritten by AACC[9]. |
|       | 7:0   | **PWRMTR_WT2_E0_HP_EVT9**  
|       |       | Default Value: 00h  
|       |       | Access: R/W Lock  
|       |       | PWRMTR_WT2_E0_HP_EVT9 This event measured by this field can be overwritten by AACC[8]. |
## PWRMTR_WT2_EEVT13TO16

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0A95Ch</td>
</tr>
</tbody>
</table>

### PWRMTR_WT2_EEVT13to16

**Description**

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>PWRMTR_WT2_E0_DP_EVT16</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT2_E0_DP_EVT16 This event measured by this field can be overwritten by AACC[15].</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Programming Notes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Must be 0 at all times.</td>
</tr>
<tr>
<td></td>
<td>23:16</td>
<td><strong>PWRMTR_WT2_E0_DP_EVT15</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT2_E0_DP_EVT15 This event measured by this field can be overwritten by AACC[14].</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Programming Notes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Must be 0 at all times.</td>
</tr>
<tr>
<td></td>
<td>15:8</td>
<td><strong>PWRMTR_WT2_E0_DP_EVT14</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT2_E0_DP_EVT14 This event measured by this field can be overwritten by AACC[13].</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Programming Notes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Must be 0 at all times.</td>
</tr>
<tr>
<td></td>
<td>7:0</td>
<td><strong>PWRMTR_WT2_E0_SP_EVT13</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT2_E0_SP_EVT13 This event measured by this field can be overwritten by AACC[12].</td>
</tr>
</tbody>
</table>
### PWRMTR_WT2_EEVT17TO20

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>PWRMTR_WT2_E0INT_EVT20</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT2_E0INT_EVT20 This event measured by this field can be overwritten by AACC[19].</td>
</tr>
<tr>
<td>23:16</td>
<td></td>
<td><strong>PWRMTR_WT2_E0INT_EVT19</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT2_E0INT_EVT19 This event measured by this field can be overwritten by AACC[18].</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Programming Notes</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Must be 0 at all times.</td>
</tr>
<tr>
<td>15:8</td>
<td></td>
<td><strong>PWRMTR_WT2_E0INT_EVT18</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT2_E0INT_EVT18 This event measured by this field can be overwritten by AACC[17].</td>
</tr>
<tr>
<td>7:0</td>
<td></td>
<td><strong>PWRMTR_WT2_E0_DP_EVT17</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT2_E0_DP_EVT17 This event measured by this field can be overwritten by AACC[16].</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Programming Notes</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Must be 0 at all times.</td>
</tr>
</tbody>
</table>
### PWRMTR_WT2_EEVT21TO24

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
<th>Programming Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>PWRMTR_WT2_E0_QP_EVT24</strong></td>
<td>00h</td>
<td>R/W Lock</td>
<td>This event measured by this field can be overwritten by AACC[23]. Must be 0 at all times.</td>
</tr>
<tr>
<td></td>
<td>23:16</td>
<td><strong>PWRMTR_WT2_E0_QP_EVT23</strong></td>
<td>00h</td>
<td>R/W Lock</td>
<td>This event measured by this field can be overwritten by AACC[22]. Must be 0 at all times.</td>
</tr>
<tr>
<td></td>
<td>15:8</td>
<td><strong>PWRMTR_WT2_E0_QP_EVT22</strong></td>
<td>00h</td>
<td>R/W Lock</td>
<td>This event measured by this field can be overwritten by AACC[21].</td>
</tr>
<tr>
<td></td>
<td>7:0</td>
<td><strong>PWRMTR_WT2_E0INT_EVT21</strong></td>
<td>00h</td>
<td>R/W Lock</td>
<td>This event measured by this field can be overwritten by AACC[20].</td>
</tr>
</tbody>
</table>
# PWRMTR_WT2_EEVT25TO28

## Register Space
- MMIO: 0/2/0

## Project
- CHV, BSW

## Source
- PRM

## Default Value
- 0x00000000

## Size (in bits)
- 32

## Address
- 0A968h

## Description

### PWRMTR_WT2_EFPU0_EVT28
- **Default Value:** 00h
- **Access:** R/W Lock
- PWRMTR_WT2_EFPU0_EVT28 This event measured by this field can be overwritten by AACC[27].

### PWRMTR_WT2_EFPU0_EVT27
- **Default Value:** 00h
- **Access:** R/W Lock
- PWRMTR_WT2_EFPU0_EVT27 This event measured by this field can be overwritten by AACC[26].

### PWRMTR_WT2_EFPU0_EVT26
- **Default Value:** 00h
- **Access:** R/W Lock
- PWRMTR_WT2_EFPU0_EVT26 This event measured by this field can be overwritten by AACC[25].

### PWRMTR_WT2_E0_QP_EVT25
- **Default Value:** 00h
- **Access:** R/W Lock
- PWRMTR_WT2_E0_QP_EVT25 This event measured by this field can be overwritten by AACC[24].

### Programming Notes
- Must be 0 at all times.
**PWRMTR_WT2_EEVT29TO32**

<table>
<thead>
<tr>
<th>PWRMTR_WT2_EEVT29TO32 - PWRMTR_WT2_EEVT29TO32</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 0A96Ch</td>
</tr>
</tbody>
</table>

PWRMTR_WT2_EEVT29to32

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>PWRMTR_WT2_E1_HP_EVT32</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT2_E1_HP_EVT32 This event measured by this field can be overwritten by AACC[31].</td>
</tr>
<tr>
<td>0</td>
<td>23:16</td>
<td><strong>PWRMTR_WT2_E1_HP_EVT31</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT2_E1_HP_EVT31 This event measured by this field can be overwritten by AACC[30].</td>
</tr>
<tr>
<td>0</td>
<td>15:8</td>
<td><strong>PWRMTR_WT2_E1_HP_EVT30</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT2_E1_HP_EVT30 This event measured by this field can be overwritten by AACC[29].</td>
</tr>
<tr>
<td>0</td>
<td>7:0</td>
<td><strong>PWRMTR_WT2_EFPU0_EVT29</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT2_EFPU0_EVT29 This event measured by this field can be overwritten by AACC[28].</td>
</tr>
</tbody>
</table>
## PWRMTR_WT2_EEVT33TO36

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>PWRMTR_WT2_E1_SP_EVT36</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT2_E1_SP_EVT36 This event measured by this field can be overwritten by AAD0[3].</td>
</tr>
<tr>
<td>0</td>
<td>23:16</td>
<td><strong>PWRMTR_WT2_E1_SP_EVT35</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT2_E1_SP_EVT35 This event measured by this field can be overwritten by AAD0[2].</td>
</tr>
<tr>
<td>0</td>
<td>15:8</td>
<td><strong>PWRMTR_WT2_E1_SP_EVT34</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT2_E1_SP_EVT34 This event measured by this field can be overwritten by AAD0[1].</td>
</tr>
<tr>
<td>0</td>
<td>7:0</td>
<td><strong>PWRMTR_WT2_E1_HP_EVT33</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT2_E1_HP_EVT33 This event measured by this field can be overwritten by AAD0[0].</td>
</tr>
</tbody>
</table>
## PWRMTR_WT2_EEVT37TO40

**Register Space:** MMIO: 0/2/0  
**Project:** CHV, BSW  
**Source:** PRM  
**Default Value:** 0x00000000  
**Size (in bits):** 32  
**Address:** 0A974h

### PWRMTR_WT2_EEVT37to40

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
</table>
| 0     | 31:24 | **PWRMTR_WT2_E1_DP_EVT40**  
|       |       | Default Value: 00h  
|       |       | Access: R/W Lock  
|       |       | PWRMTR_WT2_E1_DP_EVT40 This event measured by this field can be overwritten by AAD0[7]. |
|       | 23:16 | **PWRMTR_WT2_E1_DP_EVT39**  
|       |       | Default Value: 00h  
|       |       | Access: R/W Lock  
|       |       | PWRMTR_WT2_E1_DP_EVT39 This event measured by this field can be overwritten by AAD0[6]. |
|       | 15:8  | **PWRMTR_WT2_E1_DP_EVT38**  
|       |       | Default Value: 00h  
|       |       | Access: R/W Lock  
|       |       | PWRMTR_WT2_E1_DP_EVT38 This event measured by this field can be overwritten by AAD0[5]. |
|       | 7:0   | **PWRMTR_WT2_E1_SP_EVT37**  
|       |       | Default Value: 00h  
|       |       | Access: R/W Lock  
|       |       | PWRMTR_WT2_E1_SP_EVT37 This event measured by this field can be overwritten by AAD0[4]. |
## PWRMTR_WRT2_EEVT41TO44

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
</table>
| 0     | 31:24 | **PWRMTR_WRT2_E1INT_EVT44**  
  Default Value: 00h  
  Access: R/W Lock  
  PWRMTR_WRT2_E1INT_EVT44 This event measured by this field can be overwritten by AAD0[11]. |
|       | 23:16 | **PWRMTR_WRT2_E1INT_EVT43**  
  Default Value: 00h  
  Access: R/W Lock  
  PWRMTR_WRT2_E1INT_EVT43 This event measured by this field can be overwritten by AAD0[10]. |
|       | 15:8  | **PWRMTR_WRT2_E1INT_EVT42**  
  Default Value: 00h  
  Access: R/W Lock  
  PWRMTR_WRT2_E1INT_EVT42 This event measured by this field can be overwritten by AAD0[9]. |
|       | 7:0   | **PWRMTR_WRT2_E1_DP_EVT41**  
  Default Value: 00h  
  Access: R/W Lock  
  PWRMTR_WRT2_E1_DP_EVT41 This event measured by this field can be overwritten by AAD0[8]. |

### Programming Notes

Must be 0 at all times.
## PWRMTR_WT2_EEVT45TO48

<table>
<thead>
<tr>
<th>DWoird</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>PWRMTR_WT2_E1_QP_EVT48</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT2_E1_QP_EVT48 This event measured by this field can be overwritten by AAD0[15].</td>
</tr>
<tr>
<td>23:16</td>
<td><strong>PWRMTR_WT2_E1_QP_EVT47</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT2_E1_QP_EVT47 This event measured by this field can be overwritten by AAD0[14].</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Programming Notes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Must be 0 at all times.</td>
</tr>
<tr>
<td>15:8</td>
<td><strong>PWRMTR_WT2_E1_QP_EVT46</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT2_E1_QP_EVT46 This event measured by this field can be overwritten by AAD0[13].</td>
</tr>
<tr>
<td>7:0</td>
<td><strong>PWRMTR_WT2_E1INT_EVT45</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT2_E1INT_EVT45 This event measured by this field can be overwritten by AAD0[12].</td>
</tr>
</tbody>
</table>
# PWRMTR_WT2_EEVT49TO52

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>PWRMTR_WT2_EFPU1_EVT52</strong></td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT2_EFPU1_EVT52 This event measured by this field can be overwritten by AAD0[19].</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>23:16</td>
<td><strong>PWRMTR_WT2_EFPU1_EVT51</strong></td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT2_EFPU1_EVT51 This event measured by this field can be overwritten by AAD0[18].</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>15:8</td>
<td><strong>PWRMTR_WT2_EFPU1_EVT50</strong></td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT2_EFPU1_EVT50 This event measured by this field can be overwritten by AAD0[17].</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>7:0</td>
<td><strong>PWRMTR_WT2_E1_QP_EVT49</strong></td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT2_E1_QP_EVT49 This event measured by this field can be overwritten by AAD0[16].</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## PWRMTR_WT2_EEV53TO56

**PWRMTR_WT2_EEV53TO56 - PWRMTR_WT2_EEV53TO56**

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td>PWRMTR_WT2_EEMEM_EVT56</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT2_EEMEM_EVT56 This event measured by this field can be overwritten by AAD0[23].</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23:16</td>
<td></td>
<td>PWRMTR_WT2_EEMEM_EVT55</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT2_EEMEM_EVT55 This event measured by this field can be overwritten by AAD0[22].</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15:8</td>
<td></td>
<td>PWRMTR_WT2_EEMEM_EVT54</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT2_EEMEM_EVT54 This event measured by this field can be overwritten by AAD0[21].</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7:0</td>
<td></td>
<td>PWRMTR_WT2_EFPU1_EVT53</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT2_EFPU1_EVT53 This event measured by this field can be overwritten by AAD0[20].</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**PWRMTR_WT2_EEVT57TO58**

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td><strong>Reserved</strong></td>
<td>0000h</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15:8</td>
<td></td>
<td><strong>PWRMTR_WT2_EBUSS_EVT58</strong></td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT2_EBUSS_EVT58 This event measured by this field can be overwritten by AAD0[25].</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7:0</td>
<td></td>
<td><strong>PWRMTR_WT2_EEMEM_EVT57</strong></td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT2_EEMEM_EVT57 This event measured by this field can be overwritten by AAD0[24].</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Register Space: MMIO: 0/2/0
Project: CHV, BSW
Source: PRM
Default Value: 0x00000000
Size (in bits): 32
Address: 0A988h
# PWRMTR_WT2_REVT5TO8

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0A904h</td>
</tr>
</tbody>
</table>

## PWRMTR_WT2_REVT5to8

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>23:16</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>15:8</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>7:0</td>
<td></td>
<td><strong>PWRMTR_WT2_RFFOI_EVT5</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
</tbody>
</table>

PWRMTR_WT2_RFFOI_EVT5 This event measured by this field can be overwritten by AAC0[4].
# PWRMTR_WT2_REVT33TO36

<table>
<thead>
<tr>
<th>PWRMTR_WT2_REVT33TO36 - PWRMTR_WT2_REVT33TO36</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 0A920h</td>
</tr>
</tbody>
</table>

## PWRMTR_WT2_REVT33to36

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td>PWRMTR_WT2_RL3MI_EVT36</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT2_RL3MI_EVT36 This event measured by this field can be overwritten by AAC4[3].</td>
</tr>
<tr>
<td></td>
<td>23:16</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td>15:8</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td>7:0</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
</tbody>
</table>
## PWRMTR_WT2_REVT37TO40

### PWRMTR_WT2_REVT37TO40 - PWRMTR_WT2_REVT37TO40

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>PWRMTR_WT2_RTAOI_EVT40</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT2_RTAOI_EVT40 This event measured by this field can be overwritten by AAC4[7].</td>
</tr>
<tr>
<td>23:16</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td>15:8</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td>7:0</td>
<td></td>
<td><strong>PWRMTR_WT2_RL3MI_EVT37</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT2_RL3MI_EVT37 This event measured by this field can be overwritten by AAC4[4].</td>
</tr>
</tbody>
</table>
### PWRMTR_WT2_REVT41TO44

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0A928h</td>
</tr>
</tbody>
</table>

#### PWRMTR_WT2_REVT41to44

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td>23:16</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td>15:8</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td>7:0</td>
<td>31:24</td>
<td>PWRMTR_WT2_RTAOI_EVT41</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
</tbody>
</table>

PWRMTR_WT2_RTAOI_EVT41 This event measured by this field can be overwritten by AAC4[8].
## PWRMTR_WT2_REVT49to52

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td>PWRMTR_WT2_RTAOH_EVT52</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT2_RTAOH_EVT52 This event measured by this field can be overwritten by AAC4[19].</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>23:16</td>
<td>PWRMTR_WT2_RTAOH_EVT51</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT2_RTAOH_EVT51 This event measured by this field can be overwritten by AAC4[18].</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>15:8</td>
<td>Reserved</td>
<td>00h</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>7:0</td>
<td>PWRMTR_WT2 RTBUO_EVT49</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT2 RTBUO_EVT49 This event measured by this field can be overwritten by AAC4[16].</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### PWRMTR_WT2_REVT53TO56

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>PWRMTR_WT2_RTSO1_EVT56</strong></td>
</tr>
<tr>
<td></td>
<td>Default Value:</td>
<td>00h</td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td>PWRMTR_WT2_RTS O1_EVT56 This event measured by this field can be overwritten by AAC4[23].</td>
<td></td>
</tr>
<tr>
<td>23:16</td>
<td><strong>PWRMTR_WT2_RTACN_EVT55</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default Value:</td>
<td>00h</td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td>PWRMTR_WT2_RTACN_EVT55 This event measured by this field can be overwritten by AAC4[22].</td>
<td></td>
</tr>
<tr>
<td>15:8</td>
<td><strong>PWRMTR_WT2_RTACN_EVT54</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default Value:</td>
<td>00h</td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td>PWRMTR_WT2_RTACN_EVT54 This event measured by this field can be overwritten by AAC4[21].</td>
<td></td>
</tr>
<tr>
<td>7:0</td>
<td><strong>Reserved</strong></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td>Access: RO</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Reserved</td>
<td></td>
</tr>
</tbody>
</table>
### PWRMTR_WT2_REVT65TO68

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td>23:16</td>
<td>23:16</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td>15:8</td>
<td>PWRMTR_WT2_RGTCN_EVT66</td>
<td><strong>PWRMTR_WT2_RGTCN_EVT66</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT2_RGTCN_EVT66 This event measured by this field can be overwritten by AAC8[1].</td>
</tr>
<tr>
<td>7:0</td>
<td>PWRMTR_WT2_RGTOI_EVT65</td>
<td><strong>PWRMTR_WT2_RGTOI_EVT65</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT2_RGTOI_EVT65 This event measured by this field can be overwritten by AAC8[0].</td>
</tr>
</tbody>
</table>
### PWRMTR_WT2_REVT69TO70 - PWRMTR_WT2_REVT69TO70

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td><strong>Reserved</strong></td>
<td>0000h</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Reserved</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15:8</td>
<td>PWRMTR_WT2_RGTCN_EVT70</td>
<td>00h</td>
<td>R/W Lock</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT2_RGTCN_EVT70 This event measured by this field can be overwritten by AAC8[5].</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7:0</td>
<td>PWRMTR_WT2_RGTCN_EVT69</td>
<td>00h</td>
<td>R/W Lock</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT2_RGTCN_EVT69 This event measured by this field can be overwritten by AAC8[4].</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## PWRMTR_WT3_EEVT1to4

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>PWRMTR_WT3_EU_TH_EVT4</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT3_EU_TH_EVT4 This event measured by this field can be overwritten by AACC[3].</td>
</tr>
<tr>
<td></td>
<td>23:16</td>
<td><strong>PWRMTR_WT3_EU_GA_EVT3</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT3_EU_GA_EVT3 This event measured by this field can be overwritten by AACC[2].</td>
</tr>
<tr>
<td></td>
<td>15:8</td>
<td><strong>PWRMTR_WT3_EU_GA_EVT2</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT3_EU_GA_EVT2 This event measured by this field can be overwritten by AACC[1].</td>
</tr>
<tr>
<td></td>
<td>7:0</td>
<td><strong>PWRMTR_WT3_EU_GA_EVT1</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT3_EU_GA_EVT1 This event measured by this field can be overwritten by AACC[0].</td>
</tr>
</tbody>
</table>
## PWRMTR_WT3_EEVT5TO8

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td>PWRMTR_WT3_E0_HP_EVT8</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT3_E0_HP_EVT8 This event measured by this field can be overwritten by AACC[7].</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23:16</td>
<td></td>
<td>PWRMTR_WT3_E0_HP_EVT7</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT3_E0_HP_EVT7 This event measured by this field can be overwritten by AACC[6].</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15:8</td>
<td></td>
<td>PWRMTR_WT3_E0_HP_EVT6</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT3_E0_HP_EVT6 This event measured by this field can be overwritten by AACC[5].</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7:0</td>
<td></td>
<td>PWRMTR_WT3_EU_IO_EVT5</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT3_EU_IO_EVT5 This event measured by this field can be overwritten by AACC[4].</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## PWRMTR_WT3_EEVT9TO12

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td>PWRMTR_WT3_E0_SP_EVT12</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT3_E0_SP_EVT12 This event measured by this field can be overwritten by AACC[11].</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>23:16</td>
<td>PWRMTR_WT3_E0_SP_EVT11</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT3_E0_SP_EVT11 This event measured by this field can be overwritten by AACC[10].</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>15:8</td>
<td>PWRMTR_WT3_E0_SP_EVT10</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT3_E0_SP_EVT10 This event measured by this field can be overwritten by AACC[9].</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>7:0</td>
<td>PWRMTR_WT3_E0_HP_EVT9</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT3_E0_HP_EVT9 This event measured by this field can be overwritten by AACC[8].</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### PWRMTR_WT3_EEVT13TO16

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
<th>Programming Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>PWRMTR_WT3_E0_DP_EVT16</strong></td>
<td>00h</td>
<td>R/W Lock</td>
<td>This event measured by this field can be overwritten by AACC[15]. Must be 0 at all times.</td>
</tr>
<tr>
<td>23:16</td>
<td></td>
<td><strong>PWRMTR_WT3_E0_DP_EVT15</strong></td>
<td>00h</td>
<td>R/W Lock</td>
<td>This event measured by this field can be overwritten by AACC[14]. Must be 0 at all times.</td>
</tr>
<tr>
<td>15:8</td>
<td></td>
<td><strong>PWRMTR_WT3_E0_DP_EVT14</strong></td>
<td>00h</td>
<td>R/W Lock</td>
<td>This event measured by this field can be overwritten by AACC[13]. Must be 0 at all times.</td>
</tr>
<tr>
<td>7:0</td>
<td></td>
<td><strong>PWRMTR_WT3_E0_SP_EVT13</strong></td>
<td>00h</td>
<td>R/W Lock</td>
<td>This event measured by this field can be overwritten by AACC[12].</td>
</tr>
</tbody>
</table>
## PWRMTR_WT3_EEVT17TO20

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td>PWRMTR_WT3_E0INT_EVT20</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT3_E0INT_EVT20 This event measured by this field can be overwritten by AACC[19].</td>
</tr>
<tr>
<td>23:16</td>
<td></td>
<td>PWRMTR_WT3_E0INT_EVT19</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT3_E0INT_EVT19 This event measured by this field can be overwritten by AACC[18].</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Programming Notes</strong> Must be 0 at all times.</td>
</tr>
<tr>
<td>15:8</td>
<td></td>
<td>PWRMTR_WT3_E0INT_EVT18</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT3_E0INT_EVT18 This event measured by this field can be overwritten by AACC[17].</td>
</tr>
<tr>
<td>7:0</td>
<td></td>
<td>PWRMTR_WT3_E0_DP_EVT17</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT3_E0_DP_EVT17 This event measured by this field can be overwritten by AACC[16].</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Programming Notes</strong> Must be 0 at all times.</td>
</tr>
</tbody>
</table>
## PWRMTR_WT3_EEVT21TO24

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
</table>
| 0     | 31:24 | **PWRMTR_WT3_E0_QP_EVT24**  
Default Value: 00h  
Access: R/W Lock  
PWRMTR_WT3_E0_QP_EVT24 This event measured by this field can be overwritten by AACC[23].  
**Programming Notes**  
Must be 0 at all times. |
| 23:16 | **PWRMTR_WT3_E0_QP_EVT23**  
Default Value: 00h  
Access: R/W Lock  
PWRMTR_WT3_E0_QP_EVT23 This event measured by this field can be overwritten by AACC[22].  
**Programming Notes**  
Must be 0 at all times. |
| 15:8  | **PWRMTR_WT3_E0_QP_EVT22**  
Default Value: 00h  
Access: R/W Lock  
PWRMTR_WT3_E0_QP_EVT22 This event measured by this field can be overwritten by AACC[21]. |
| 7:0   | **PWRMTR_WT3_E0INT_EVT21**  
Default Value: 00h  
Access: R/W Lock  
PWRMTR_WT3_E0INT_EVT21 This event measured by this field can be overwritten by AACC[20]. |
### PWRMTR_WT3_EEVT25TO28

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td>PWRMTR_WT3_EFPU0_EVT28</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT3_EFPU0_EVT28 This event measured by this field can be overwritten by AACC[27].</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23:16</td>
<td></td>
<td>PWRMTR_WT3_EFPU0_EVT27</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT3_EFPU0_EVT27 This event measured by this field can be overwritten by AACC[26].</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15:8</td>
<td></td>
<td>PWRMTR_WT3_EFPU0_EVT26</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT3_EFPU0_EVT26 This event measured by this field can be overwritten by AACC[25].</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7:0</td>
<td></td>
<td>PWRMTR_WT3_E0_QP_EVT25</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT3_E0_QP_EVT25 This event measured by this field can be overwritten by AACC[24].</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Programming Notes**

Must be 0 at all times.
## PWRMTR_WT3_EEVT29TO32

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td>PWRMTR_WT3_E1_HP_EVT32</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT3_E1_HP_EVT32 This event measured by this field can be overwritten by AACC[31].</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23:16</td>
<td></td>
<td>PWRMTR_WT3_E1_HP_EVT31</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT3_E1_HP_EVT31 This event measured by this field can be overwritten by AACC[30].</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15:8</td>
<td></td>
<td>PWRMTR_WT3_E1_HP_EVT30</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT3_E1_HP_EVT30 This event measured by this field can be overwritten by AACC[29].</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7:0</td>
<td></td>
<td>PWRMTR_WT3_EFPU0_EVT29</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT3_EFPU0_EVT29 This event measured by this field can be overwritten by AACC[28].</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
PWRMTR_WT3_EEVT33TO36

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td>PWRMTR_WT3_E1_SP_EVT36</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT3_E1_SP_EVT36 This event measured by this field can be overwritten by AAD0[3].</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>23:16</td>
<td>PWRMTR_WT3_E1_SP_EVT35</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT3_E1_SP_EVT35 This event measured by this field can be overwritten by AAD0[2].</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>15:8</td>
<td>PWRMTR_WT3_E1_SP_EVT34</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT3_E1_SP_EVT34 This event measured by this field can be overwritten by AAD0[1].</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>7:0</td>
<td>PWRMTR_WT3_E1_HP_EVT33</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT3_E1_HP_EVT33 This event measured by this field can be overwritten by AAD0[0].</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## PWRMTR_WT3_EEVT37TO40

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>31:24</td>
<td><strong>PWRMTR_WT3_E1_DP_EVT40</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT3_E1_DP_EVT40 This event measured by this field can be overwritten by AAD0[7].</td>
</tr>
<tr>
<td></td>
<td>23:16</td>
<td><strong>PWRMTR_WT3_E1_DP_EVT39</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT3_E1_DP_EVT39 This event measured by this field can be overwritten by AAD0[6].</td>
</tr>
<tr>
<td></td>
<td>15:8</td>
<td><strong>PWRMTR_WT3_E1_DP_EVT38</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT3_E1_DP_EVT38 This event measured by this field can be overwritten by AAD0[5].</td>
</tr>
<tr>
<td></td>
<td>7:0</td>
<td><strong>PWRMTR_WT3_E1_SP_EVT37</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT3_E1_SP_EVT37 This event measured by this field can be overwritten by AAD0[4].</td>
</tr>
</tbody>
</table>
## PWRMTR_WT3_EEVT41TO44

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td>PWRMTR_WT3_E1INT_EVT44</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT3_E1INT_EVT44 This event measured by this field can be overwritten by AAD0[11].</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>23:16</td>
<td>PWRMTR_WT3_E1INT_EVT43</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT3_E1INT_EVT43 This event measured by this field can be overwritten by AAD0[10].</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>15:8</td>
<td>PWRMTR_WT3_E1INT_EVT42</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT3_E1INT_EVT42 This event measured by this field can be overwritten by AAD0[9].</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>7:0</td>
<td>PWRMTR_WT3_E1_DP_EVT41</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT3_E1_DP_EVT41 This event measured by this field can be overwritten by AAD0[8].</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Programming Notes**

Must be 0 at all times.
### Command Reference: Registers

#### PWRMTR_WT3_EEVT45TO48

<table>
<thead>
<tr>
<th>Register Space</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits)</td>
<td>32</td>
</tr>
<tr>
<td>Address</td>
<td>0AA7Ch</td>
</tr>
</tbody>
</table>

**PWRMTR_WT3_EEVT45to48**

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>PWRMTR_WT3_E1_QP_EVT48</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT3_E1_QP_EVT48 This event measured by this field can be overwritten by AAD0[15].</td>
</tr>
</tbody>
</table>

| 23:16 | **PWRMTR_WT3_E1_QP_EVT47**  |
|       | Default Value: 00h             |
|       | Access: R/W Lock                |
|       | PWRMTR_WT3_E1_QP_EVT47 This event measured by this field can be overwritten by AAD0[14]. |

**Programming Notes**

Must be 0 at all times.

| 15:8 | **PWRMTR_WT3_E1_QP_EVT46**  |
|      | Default Value: 00h             |
|      | Access: R/W Lock                |
|      | PWRMTR_WT3_E1_QP_EVT46 This event measured by this field can be overwritten by AAD0[13]. |

| 7:0 | **PWRMTR_WT3_E1INT_EVT45**  |
|     | Default Value: 00h             |
|     | Access: R/W Lock                |
|     | PWRMTR_WT3_E1INT_EVT45 This event measured by this field can be overwritten by AAD0[12]. |
## PWRMTR_WT3_EEVT49TO52

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>PWRMTR_WT3_EFPU1_EVT52</strong></td>
<td>00h</td>
<td>R/W Lock</td>
<td>This event measured by this field can be overwritten by AAD0[19].</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23:16</td>
<td></td>
<td><strong>PWRMTR_WT3_EFPU1_EVT51</strong></td>
<td>00h</td>
<td>R/W Lock</td>
<td>This event measured by this field can be overwritten by AAD0[18].</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15:8</td>
<td></td>
<td><strong>PWRMTR_WT3_EFPU1_EVT50</strong></td>
<td>00h</td>
<td>R/W Lock</td>
<td>This event measured by this field can be overwritten by AAD0[17].</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7:0</td>
<td></td>
<td><strong>PWRMTR_WT3_E1_QP_EVT49</strong></td>
<td>00h</td>
<td>R/W Lock</td>
<td>This event measured by this field can be overwritten by AAD0[16].</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access:</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### PWRMTR_WT3_EEVT53TO56

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td>PWRMTR_WT3_EEMEM_EVT56</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>23:16</td>
<td>PWRMTR_WT3_EEMEM_EVT55</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>15:8</td>
<td>PWRMTR_WT3_EEMEM_EVT54</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>7:0</td>
<td>PWRMTR_WT3_EFPU1_EVT53</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
</tbody>
</table>

PWRMTR_WT3_EEVT53to56

**Register Space:** MMIO: 0/2/0

**Project:** CHV, BSW

**Source:** PRM

**Default Value:** 0x00000000

**Size (in bits):** 32

**Address:** 0AA84h

PWRMTR_WT3_EEVT53to56 This event measured by this field can be overwritten by AAD0[23].

PWRMTR_WT3_EEVT55 This event measured by this field can be overwritten by AAD0[22].

PWRMTR_WT3_EEVT54 This event measured by this field can be overwritten by AAD0[21].

PWRMTR_WT3_EEVT53 This event measured by this field can be overwritten by AAD0[20].
PWRMTR_WT3_EEVT57TO58

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td><strong>Reserved</strong></td>
<td>0000h</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>PWRMTR_WT3_EBUSS_EVT58</strong></td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT3_EBUSS_EVT58 This event measured by this field can be overwritten by AAD0[25].</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7:0</td>
<td></td>
<td><strong>PWRMTR_WT3_EEMEM_EVT57</strong></td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT3_EEMEM_EVT57 This event measured by this field can be overwritten by AAD0[24].</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## PWRMTR_WT3_REVT5TO8

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td>Reserved</td>
<td>00h</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Reserved</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Reserved</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Reserved</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>7:0</td>
<td>PWRMTR_WT3_RFFOI_EVT5</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>PWRMTR_WT3_RFFOI_EVT5</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Register Space:** MMIO: 0/2/0
- **Project:** CHV, BSW
- **Source:** PRM
- **Default Value:** 0x00000000
- **Size (in bits):** 32
- **Address:** 0AA04h

---

This event measured by this field can be overwritten by AAC0[4].
### PWRMTR_WT3_REVT33TO36

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0AA20h</td>
</tr>
</tbody>
</table>

#### PWRMTR_WT3_REVT33to36

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>PWRMTR_WT3_RL3MI_EVT36</strong></td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td>23:16</td>
<td><strong>Reserved</strong></td>
<td>00h</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td>15:8</td>
<td><strong>Reserved</strong></td>
<td>00h</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td>7:0</td>
<td><strong>Reserved</strong></td>
<td>00h</td>
<td>RO</td>
</tr>
</tbody>
</table>
**PWRMTR_WT3_REVT37TO40**

<table>
<thead>
<tr>
<th>PWRMTR_WT3_REVT37TO40 - PWRMTR_WT3_REVT37TO40</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space</strong>: MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project</strong>: CHV, BSW</td>
</tr>
<tr>
<td><strong>Source</strong>: PRM</td>
</tr>
<tr>
<td><strong>Default Value</strong>: 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits)</strong>: 32</td>
</tr>
<tr>
<td><strong>Address</strong>: 0AA24h</td>
</tr>
</tbody>
</table>

**PWRMTR_WT3_REVT37to40**

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td>PWRMTR_WT3_RTAOI_EVT40</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT3_RTAOI_EVT40 This event measured by this field can be overwritten by AAC4[7].</td>
</tr>
<tr>
<td>23:16</td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td>15:8</td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td>7:0</td>
<td></td>
<td>PWRMTR_WT3_RL3MI_EVT37</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT3_RL3MI_EVT37 This event measured by this field can be overwritten by AAC4[4].</td>
</tr>
</tbody>
</table>
### PWRMTR_WT3_REV41TO44 - PWRMTR_WT3_REV41TO44

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>31:24</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td>23:16</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td>15:8</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td>7:0</td>
<td><strong>PWRMTR_WT3_RTAOI_EVT41</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT3_RTAOI_EVT41 This event measured by this field can be overwritten by AAC4[8].</td>
</tr>
</tbody>
</table>
### PWRMTR_WT3_REVT49TO52

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td>PWRMTR_WT3_RTAOH_EVT52</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT3_RTAOH_EVT52 This event measured by this field can be overwritten by AAC4[19].</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23:16</td>
<td></td>
<td>PWRMTR_WT3_RTAOH_EVT51</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT3_RTAOH_EVT51 This event measured by this field can be overwritten by AAC4[18].</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15:8</td>
<td></td>
<td>Reserved</td>
<td>00h</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7:0</td>
<td></td>
<td>PWRMTR_WT3_RTAOI_EVT49</td>
<td>00h</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT3_RTAOI_EVT49 This event measured by this field can be overwritten by AAC4[16].</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## PWRMTR_WT3_REVT53TO56

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>PWRMTR_WT3_RTSO1_EVT56</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT3_RTSO1_EVT56 This event measured by this field can be overwritten by AAC4[23].</td>
</tr>
<tr>
<td></td>
<td>23:16</td>
<td><strong>PWRMTR_WT3_RTACN_EVT55</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT3_RTACN_EVT55 This event measured by this field can be overwritten by AAC4[22].</td>
</tr>
<tr>
<td></td>
<td>15:8</td>
<td><strong>PWRMTR_WT3_RTACN_EVT54</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWRMTR_WT3_RTACN_EVT54 This event measured by this field can be overwritten by AAC4[21].</td>
</tr>
<tr>
<td></td>
<td>7:0</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
</tbody>
</table>
### PWRMTR_WT3_REVT65TO68 - PWRMTR_WT3_REVT65TO68

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td>Default Value:</td>
<td>00h</td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td></td>
<td>23:16</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td>Default Value:</td>
<td>00h</td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>15:8</td>
<td></td>
<td><strong>PWRMTR_WT3_RGTCN_EVT66</strong></td>
</tr>
<tr>
<td></td>
<td>Default Value:</td>
<td>00h</td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td>PWRMTR_WT3_RGTCN_EVT66 This event measured by this field can be overwritten by AAC8[1].</td>
<td></td>
</tr>
<tr>
<td>7:0</td>
<td></td>
<td><strong>PWRMTR_WT3_RGTOI_EVT65</strong></td>
</tr>
<tr>
<td></td>
<td>Default Value:</td>
<td>00h</td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td>PWRMTR_WT3_RGTOI_EVT65 This event measured by this field can be overwritten by AAC8[0].</td>
<td></td>
</tr>
</tbody>
</table>

Register Space: MMIO: 0/2/0
Project: CHV, BSW
Source: PRM
Default Value: 0x00000000
Size (in bits): 32
Address: 0AA40h

**PWRMTR_WT3_REVT65to68**
PWRMTR_WT3_REVT69TO70

<table>
<thead>
<tr>
<th>PWRMTR_WT3_REVT69TO70 - PWRMTR_WT3_REVT69TO70</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 0AA44h</td>
</tr>
</tbody>
</table>

PWRMTR_WT3_REVT69to70

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td>Default Value: 0000h</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access: RO</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>15:8</td>
<td>PWRMTR_WT3_RGTCN_EVT70</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default Value: 00h</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access: R/W Lock</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PWRMTR_WT3_RGTCN_EVT70 This event measured by this field can be overwritten by AAC8[5].</td>
<td></td>
</tr>
<tr>
<td>7:0</td>
<td>PWRMTR_WT3_RGTCN_EVT69</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default Value: 00h</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access: R/W Lock</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PWRMTR_WT3_RGTCN_EVT69 This event measured by this field can be overwritten by AAC8[4].</td>
<td></td>
</tr>
</tbody>
</table>
# PWRMTREVTORE0

<table>
<thead>
<tr>
<th>PWRMTREVTORE0 - PWRMTREVTORE0</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong></td>
</tr>
<tr>
<td>MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong></td>
</tr>
<tr>
<td>CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong></td>
</tr>
<tr>
<td>PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong></td>
</tr>
<tr>
<td>0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong></td>
</tr>
<tr>
<td>32</td>
</tr>
<tr>
<td><strong>Address:</strong></td>
</tr>
<tr>
<td>0AACCh</td>
</tr>
</tbody>
</table>

**GT_CR_POWER_METER_EVENT OVERRIDE_EU1_32**

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>PMEVORE0</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
</tbody>
</table>

Power Meter Event Override for Render Events. Allows user to override signal coming in from Gfx engine and count event every cycle on a per event basis.

For each event:
- 0: Power Meter works normally with event input taken from Gfx engine.
- 1: Force event input to weighting logic high every clock, overriding event data coming in from Gfx engine.

This register is the override for events 1 through 32, where bit 0 corresponds to event 1, and bit 31 corresponds to event 32.
## PWRMTREVTORE1

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:26</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td>25:0</td>
<td>PMEVORE1</td>
<td>Power Meter Event Override for Render Events. Allows user to override signal coming in from Gfx engine and count event every cycle on a per event basis. For each event: 0: Power Meter works normally with event input taken from Gfx engine. 1: Force event input to weighting logic high every cuclk, overriding event data coming in from Gfx engine. This register is the override for events 33 through 64, where bit 0 corresponds to event 33, and bit 25 corresponds to event 58.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
</tbody>
</table>
# PWRMTREVTORM0

## PWRMTREVTORM0 - PWRMTREVTORM0

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0AAD4h</td>
</tr>
</tbody>
</table>

**GT_CR_POWER_METER_EVENT_OVERRIDE_MEDIA1_32**

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>PMEVORM0</strong></td>
</tr>
</tbody>
</table>

- **Default Value:** 00000000h
- **Access:** R/W Lock

Power Meter Event Override for Media Events. Allows user to override signal coming in from Gfx engine and count event every cycle on a per event basis.

For each event:
- 0: Power Meter works normally with event input taken from Gfx engine.
- 1: Force event input to weighting logic high every clock, overriding event data coming in from Gfx engine.

This register is the override for events 1 through 32, where bit 0 corresponds to event 1, and bit 31 corresponds to event 32."
## PWRMTREVTORM1

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:4</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td>3:0</td>
<td>PMEVORM1</td>
<td>Power Meter Event Override for Media Events. Allows user to override signal coming in from Gfx engine and count event every cycle on a per event basis. For each event: 0: Power Meter works normally with event input taken from Gfx engine. 1: Force event input to weighting logic high every clock, overriding event data coming in from Gfx engine. This register is the override for events 33 through 36, where bit 0 corresponds to event 33, and bit 3 corresponds to event 36.</td>
</tr>
</tbody>
</table>

**Register Space:** MMIO: 0/2/0  
**Project:** CHV, BSW  
**Source:** PRM  
**Default Value:** 0x00000000  
**Size (in bits):** 32  
**Address:** 0AAD8h
## PWRMTREVTORR0

<table>
<thead>
<tr>
<th><strong>PWRMTREVTORR0 - PWRMTREVTORR0</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 0AAC0h</td>
</tr>
</tbody>
</table>

### GT_CR_POWER_METER_EVENT_OVERRIDE_RENDER1_32

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>PMEVORR0</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W Lock</td>
</tr>
</tbody>
</table>

Power Meter Event Override for Render Events. Allows user to override signal coming in from Gfx engine and count event every cycle on a per event basis.

For each event:

- **0:** Power Meter works normally with event input taken from Gfx engine.
- **1:** Force event input to weighting logic high every clock, overriding event data coming in from Gfx engine.

This register is the override for events 1 through 32, where bit 0 corresponds to event 1, and bit 31 corresponds to event 32.
# PWRMTREVTORR1

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0AAC4h</td>
</tr>
</tbody>
</table>

**GT_CR_POWER_METER_EVENT_OVERRIDE_RENDER33_64**

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>PMEVORR1</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W Lock</td>
</tr>
</tbody>
</table>

Power Meter Event Override for Render Events. Allows user to override signal coming in from Gfx engine and count event every cycle on a per event basis.

For each event:

- **0**: Power Meter works normally with event input taken from Gfx engine.
- **1**: Force event input to weighting logic high every clock, overriding event data coming in from Gfx engine.

This register is the override for events 33 through 64, where bit 0 corresponds to event 33, and bit 31 corresponds to event 64.
## PWRMTREVTORR2

### Register Details

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space</td>
<td>MMIO: 0/2/0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Project</td>
<td>CHV, BSW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Source</td>
<td>PRM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default Value</td>
<td>0x00000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Size (in bits)</td>
<td>32</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Address</td>
<td>0AAC8h</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GT_CR_POWER_METER_EVENT_OVERRIDE_RENDER65_70</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:6</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td>5:0</td>
<td></td>
<td><strong>PMEVORR2</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Lock</td>
</tr>
</tbody>
</table>

Power Meter Event Override for Render Events. Allows user to override signal coming in from Gfx engine and count event every cycle on a per event basis.

For each event:
- 0: Power Meter works normally with event input taken from Gfx engine.
- 1: Force event input to weighting logic high every clock, overriding event data coming in from Gfx engine.

This register is the override for events 65 through 70, where bit 0 corresponds to event 65, and bit 5 corresponds to event 70.
### RAM Clock Gating Control 1

**RCGCTL1 - RAM Clock Gating Control 1**

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31</td>
<td>USBunit RAM Clock Gating Disable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>USBunit RAM Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>30</td>
<td></td>
<td>VLFunit RAM Clock Gating Disable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VLFunit RAM Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>29</td>
<td></td>
<td>VISunit RAM Clock Gating Disable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VISunit RAM Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>28</td>
<td></td>
<td>STCunit RAM Clock Gating Disable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>STCunit RAM Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>Register</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>-------------------</td>
<td>------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>RCGCTRL1</td>
<td>RAM Clock Gating Control 1</td>
<td></td>
</tr>
<tr>
<td>27 TDSunit</td>
<td>Access: R/W</td>
<td></td>
</tr>
<tr>
<td>RAM Clock Gating</td>
<td>TDSunit RAM Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td>Disable</td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they</td>
<td></td>
</tr>
<tr>
<td></td>
<td>are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
<tr>
<td>26 VMCunit</td>
<td>Access: R/W</td>
<td></td>
</tr>
<tr>
<td>RAM Clock Gating</td>
<td>VMCunit RAM Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td>Disable</td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they</td>
<td></td>
</tr>
<tr>
<td></td>
<td>are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
<tr>
<td>25 QRCunit</td>
<td>Access: R/W</td>
<td></td>
</tr>
<tr>
<td>RAM Clock Gating</td>
<td>QRCunit RAM Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td>Disable</td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they</td>
<td></td>
</tr>
<tr>
<td></td>
<td>are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
<tr>
<td>24 SCunit</td>
<td>Access: R/W</td>
<td></td>
</tr>
<tr>
<td>RAM Clock Gating</td>
<td>SCunit RAM Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td>Disable</td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they</td>
<td></td>
</tr>
<tr>
<td></td>
<td>are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
<tr>
<td>23 SVLunit</td>
<td>Access: R/W</td>
<td></td>
</tr>
<tr>
<td>RAM Clock Gating</td>
<td>SVLunit RAM Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td>Disable</td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they</td>
<td></td>
</tr>
<tr>
<td></td>
<td>are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
<tr>
<td>22 VFunit</td>
<td>Access: R/W</td>
<td></td>
</tr>
<tr>
<td>RAM Clock Gating</td>
<td>VFunit RAM Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td>Disable</td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they</td>
<td></td>
</tr>
<tr>
<td></td>
<td>are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
</tbody>
</table>
## RCGCTL1 - RAM Clock Gating Control 1

<table>
<thead>
<tr>
<th>Register Name</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>21 URBunit RAM Clock Gating Disable</strong></td>
<td>R/W</td>
<td>Access: R/W&lt;br&gt;URBunit RAM Clock Gating Disable Control:&lt;br&gt;'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)&lt;br&gt;'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td><strong>20 GAMWunit RAM Clock Gating Disable</strong></td>
<td>R/W</td>
<td>Access: R/W&lt;br&gt;GAMWunit RAM Clock Gating Disable Control:&lt;br&gt;'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)&lt;br&gt;'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td><strong>19 SVGunit RAM Clock Gating Disable</strong></td>
<td>R/W</td>
<td>Access: R/W&lt;br&gt;SVGunit RAM Clock Gating Disable Control:&lt;br&gt;'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)&lt;br&gt;'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td><strong>18 RCZunit RAM Clock Gating Disable</strong></td>
<td>R/W</td>
<td>Access: R/W&lt;br&gt;RCZunit RAM Clock Gating Disable Control:&lt;br&gt;'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)&lt;br&gt;'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td><strong>17 RCPBEunit RAM Clock Gating Disable</strong></td>
<td>R/W</td>
<td>Access: R/W&lt;br&gt;RCPBEunit RAM Clock Gating Disable Control:&lt;br&gt;'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)&lt;br&gt;'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td><strong>16 RCCunit RAM Clock Gating Disable</strong></td>
<td>R/W</td>
<td>Access: R/W&lt;br&gt;RCCunit RAM Clock Gating Disable Control:&lt;br&gt;'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)&lt;br&gt;'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>
### RCGCTRL1 - RAM Clock Gating Control 1

<table>
<thead>
<tr>
<th></th>
<th>PSDunit RAM Clock Gating Disable</th>
<th></th>
<th>MTunit RAM Clock Gating Disable</th>
<th></th>
<th>SBEunit RAM Clock Gating Disable</th>
<th></th>
<th>IZunit RAM Clock Gating Disable</th>
<th></th>
<th>Reserved</th>
<th></th>
<th>ICunit RAM Clock Gating Disable</th>
<th></th>
<th>HIZunit RAM Clock Gating Disable</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td><strong>Access:</strong> R/W</td>
<td></td>
<td><strong>Access:</strong> R/W</td>
<td></td>
<td><strong>Access:</strong> R/W</td>
<td></td>
<td><strong>Access:</strong> R/W</td>
<td></td>
<td><strong>Access:</strong> R/W</td>
<td></td>
<td><strong>Access:</strong> R/W</td>
<td></td>
<td><strong>Access:</strong> R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>PSDunit RAM Clock Gating Disable Control:</strong></td>
<td></td>
<td><strong>MTunit RAM Clock Gating Disable Control:</strong></td>
<td></td>
<td><strong>SBEunit RAM Clock Gating Disable Control:</strong></td>
<td></td>
<td><strong>IZunit RAM Clock Gating Disable Control:</strong></td>
<td></td>
<td><strong>Reserved</strong></td>
<td></td>
<td><strong>ICunit RAM Clock Gating Disable Control:</strong></td>
<td></td>
<td><strong>HIZunit RAM Clock Gating Disable Control:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
</tbody>
</table>
## RCGCTRL1 - RAM Clock Gating Control 1

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
</table>
| **GAMunit RAM Clock Gating Disable** | Default Value: 0b  
Access: R/W  
GAMunit RAM Clock Gating Disable Control:  
'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
'1': Clock Gating Disabled. (i.e., clocks are toggling, always) |               |         |
| **BCunit RAM Clock Gating Disable** | Access: R/W  
BCunit RAM Clock Gating Disable Control:  
'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
'1': Clock Gating Disabled. (i.e., clocks are toggling, always) |               |         |
| **HDCunit RAM Clock Gating Disable** | Access: R/W  
GAFSunit RAM Clock Gating Disable Control:  
'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
'1': Clock Gating Disabled. (i.e., clocks are toggling, always) |               |         |
| **DMunit RAM Clock Gating Disable** | Access: R/W  
DMunit RAM Clock Gating Disable Control:  
'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
'1': Clock Gating Disabled. (i.e., clocks are toggling, always) |               |         |
| **WMFEunit RAM Clock Gating Disable** | Access: R/W  
WMFEunit RAM Clock Gating Disable Control:  
'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
'1': Clock Gating Disabled. (i.e., clocks are toggling, always) |               |         |
| **CSunit RAM Clock Gating Disable** | Access: R/W  
CSunit RAM Clock Gating Disable Control:  
'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
'1': Clock Gating Disabled. (i.e., clocks are toggling, always) |               |         |
### RCGCTL1 - RAM Clock Gating Control 1

<table>
<thead>
<tr>
<th></th>
<th><strong>BLBunit RAM Clock Gating Disable</strong></th>
<th>Access: R/W</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>BLBunit RAM Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th><strong>MPCunit RAM Clock Gating Disable</strong></th>
<th>Access: R/W</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>MPCunit RAM Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th><strong>BFunit RAM Clock Gating Disable</strong></th>
<th>Access: R/W</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>BFunit RAM Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
</tbody>
</table>
## RAM Clock Gating Control 1

### RCGCTL1 - RAM Clock Gating Control 1

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000100</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>09410h</td>
</tr>
</tbody>
</table>

### RAM Clock Gating Control Registers.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31</td>
<td>USBunit RAM Clock Gating Disable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>USBunit RAM Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>30</td>
<td></td>
<td>VLFunit RAM Clock Gating Disable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VLFunit RAM Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>29</td>
<td></td>
<td>VISunit RAM Clock Gating Disable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VISunit RAM Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>28</td>
<td></td>
<td>STCunit RAM Clock Gating Disable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>STCunit RAM Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>
## RCGCTL1 - RAM Clock Gating Control 1

<table>
<thead>
<tr>
<th>Register</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
</table>
| TDSunit RAM Clock Gating Disable | R/W    | TDSunit RAM Clock Gating Disable Control:  
|                   |        | '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
|                   |        | '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)            |
| VMCunit RAM Clock Gating Disable | R/W    | VMCunit RAM Clock Gating Disable Control:  
|                   |        | '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
|                   |        | '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)            |
| QRCunit RAM Clock Gating Disable | R/W    | QRCunit RAM Clock Gating Disable Control:  
|                   |        | '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
|                   |        | '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)            |
| SCunit RAM Clock Gating Disable | R/W    | SCunit RAM Clock Gating Disable Control:  
|                   |        | '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
|                   |        | '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)            |
| SVLunit RAM Clock Gating Disable | R/W    | SVLunit RAM Clock Gating Disable Control:  
|                   |        | '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
|                   |        | '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)            |
| VFunit RAM Clock Gating Disable | R/W    | VFunit RAM Clock Gating Disable Control:  
|                   |        | '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
|                   |        | '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)            |
## RCGCTL1 - RAM Clock Gating Control 1

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
<th>Access</th>
<th>Control Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>21</td>
<td><strong>URBunit RAM Clock Gating Disable</strong></td>
<td>R/W</td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>20</td>
<td><strong>GAMWunit RAM Clock Gating Disable</strong></td>
<td>R/W</td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>19</td>
<td><strong>SVGunit RAM Clock Gating Disable</strong></td>
<td>R/W</td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>18</td>
<td><strong>RCZunit RAM Clock Gating Disable</strong></td>
<td>R/W</td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>17</td>
<td><strong>RCPBEunit RAM Clock Gating Disable</strong></td>
<td>R/W</td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>16</td>
<td><strong>RCCunit RAM Clock Gating Disable</strong></td>
<td>R/W</td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>
### RCGCTL1 - RAM Clock Gating Control 1

<table>
<thead>
<tr>
<th>Register</th>
<th>Function</th>
<th>Access</th>
<th>Control Description</th>
</tr>
</thead>
</table>
| 15       | **PSDunit RAM Clock Gating Disable**          | R/W    | '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
          |       | '1' : Clock Gating Disabled. (i.e., clocks are toggling, always) |
| 14       | **MTunit RAM Clock Gating Disable**           | R/W    | '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
          |       | '1' : Clock Gating Disabled. (i.e., clocks are toggling, always) |
| 13       | **SBEunit RAM Clock Gating Disable**          | R/W    | '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
          |       | '1' : Clock Gating Disabled. (i.e., clocks are toggling, always) |
| 12       | **IZunit RAM Clock Gating Disable**           | R/W    | '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
          |       | '1' : Clock Gating Disabled. (i.e., clocks are toggling, always) |
| 11       | **Reserved**                                  | R/W    |                                                          |
| 10       | **ICunit RAM Clock Gating Disable**           | R/W    | '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
          |       | '1' : Clock Gating Disabled. (i.e., clocks are toggling, always) |
| 9        | **HIZunit RAM Clock Gating Disable**          | R/W    | '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
<pre><code>      |       | '1' : Clock Gating Disabled. (i.e., clocks are toggling, always) |
</code></pre>
<table>
<thead>
<tr>
<th>Register Name</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>RCGCTL1 - RAM Clock Gating Control 1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| 8  GAMunit RAM Clock Gating Disable | Default Value: 1b  
Access: R/W  
GAMunit RAM Clock Gating Disable Control:  
'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
'1': Clock Gating Disabled. (i.e., clocks are toggling, always) |               |         |
| 7  BCunit RAM Clock Gating Disable | Access: R/W  
BCunit RAM Clock Gating Disable Control:  
'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
'1': Clock Gating Disabled. (i.e., clocks are toggling, always) |               |         |
| 6  HDCunit RAM Clock Gating Disable | Access: R/W  
HDCunit RAM Clock Gating Disable Control:  
'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
'1': Clock Gating Disabled. (i.e., clocks are toggling, always) |               |         |
| 5  DMunit RAM Clock Gating Disable | Access: R/W  
DMunit RAM Clock Gating Disable Control:  
'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
'1': Clock Gating Disabled. (i.e., clocks are toggling, always) |               |         |
| 4  WMFEunit RAM Clock Gating Disable | Access: R/W  
WMFEunit RAM Clock Gating Disable Control:  
'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
'1': Clock Gating Disabled. (i.e., clocks are toggling, always) |               |         |
| 3  CSunit RAM Clock Gating Disable | Access: R/W  
CSunit RAM Clock Gating Disable Control:  
'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
'1': Clock Gating Disabled. (i.e., clocks are toggling, always) |               |         |
### RCGCTL1 - RAM Clock Gating Control 1

<table>
<thead>
<tr>
<th></th>
<th>BLBunit RAM Clock Gating Disable</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td><strong>Access:</strong> R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>BLBunit RAM Clock Gating Disable Control:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>MPCunit RAM Clock Gating Disable</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><strong>Access:</strong> R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>MPCunit RAM Clock Gating Disable Control:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>BFunit RAM Clock Gating Disable</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td><strong>Access:</strong> R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>BFunit RAM Clock Gating Disable Control:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
</tbody>
</table>
## RAM Clock Gating Control 2

### RCGCTL2 - RAM Clock Gating Control 2

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x7FC00000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>09414h</td>
</tr>
</tbody>
</table>

### RAM Clock Gating Control Registers.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31</td>
<td><strong>SPARE 2 clock gate disable</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>SPARE 2 unit Clock Gating Disable Control:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>30:28</td>
<td></td>
<td><strong>VMCRunit clock gate disable</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 111b</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>VMCR unit Clock Gating Disable Control:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>27:25</td>
<td></td>
<td><strong>SMCRunit clock gate disable</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 111b</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>SMCR unit Clock Gating Disable Control:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>24:22</td>
<td></td>
<td><strong>MCRunit clock gate disable</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 111b</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>MCR unit Clock Gating Disable Control:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td><strong>RCGCTL2 - RAM Clock Gating Control 2</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td><strong>Reserved</strong></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td><strong>WVIS unit RAM clock gate disable</strong></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td>WVIS unit RAM Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td><strong>WAVM unit RAM clock gate disable</strong></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td>WAVM unit RAM Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td><strong>WHME unit RAM clock gate disable bit</strong></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td>WHME unit RAM Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td><strong>WIME unit RAM clock gate disable</strong></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td>WIME unit RAM Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td><strong>WMPC unit RAM clock gating disable</strong></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td>WMPC unit RAM Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td><strong>SDE unit RAM clock gate disable</strong></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td>SDE unit RAM Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
</tbody>
</table>
### RCGCTRL2 - RAM Clock Gating Control 2

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
<th>Access</th>
<th>Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>VSHM unit clock gate disable</td>
<td>R/W</td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>13</td>
<td>DAPRTS unit RAM clock gate disable</td>
<td>R/W</td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1' : Clock Gating Disabled.</td>
</tr>
<tr>
<td>12</td>
<td>GS unit RAM clock gate disable</td>
<td>R/W</td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>11</td>
<td>Reserved</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>GAMT unit RAM clock gate disable bit</td>
<td>R/W</td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>9</td>
<td>VCW unit RAM clock gate disable</td>
<td>R/W</td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>8</td>
<td>VEO unit RAM clock gate disable</td>
<td>R/W</td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>
## RCGCTL2 - RAM Clock Gating Control 2

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
<th>Access</th>
<th>Control Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>IMEunit RAM clock gate disable</td>
<td>R/W</td>
<td>Clock Gating Disable Control: &lt;br&gt; '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) &lt;br&gt; '1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>6</td>
<td>CREunit RAM clock gate disable</td>
<td>R/W</td>
<td>Clock Gating Disable Control: &lt;br&gt; '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) &lt;br&gt; '1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>5</td>
<td>RSunit RAM clock gate disable</td>
<td>R/W</td>
<td>Clock Gating Disable Control: &lt;br&gt; '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) &lt;br&gt; '1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>4</td>
<td>MSCunit RAM clock Gating Disable</td>
<td>R/W</td>
<td>Clock Gating Disable Control: &lt;br&gt; '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) &lt;br&gt; '1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>3</td>
<td>VMXunit RAM clock Gating Disable</td>
<td>R/W</td>
<td>Clock Gating Disable Control: &lt;br&gt; '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) &lt;br&gt; '1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>2</td>
<td>GAunit RAM Clock Gating Disable for all EUs</td>
<td>R/W</td>
<td>Clock Gating Disable Control For all EUs in each Row: &lt;br&gt; '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) &lt;br&gt; '1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>
# RCGCTL2 - RAM Clock Gating Control 2

<table>
<thead>
<tr>
<th>1</th>
<th><strong>VSunit RAM Clock Gating Disable</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td>VSunit RAM Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>0</th>
<th><strong>HSunit RAM Clock Gating Disable</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td>HSunit RAM Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>
## RAM Clock Gating Control 2

**RCGCTL2 - RAM Clock Gating Control 2**

<table>
<thead>
<tr>
<th>Register Space: MMIO: 0/2/0</th>
<th>Project: CHV, BSW</th>
<th>Source: PRM</th>
<th>Default Value: 0xFFC00000</th>
<th>Size (in bits): 32</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address: 09414h</td>
<td>RAM Clock Gating Control Registers.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31</td>
<td><strong>1x2X Assign fub XOR clock gate disable</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 1b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>XOR based unit level clock gating disable in 1x2x_asgn fub:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0': XOR Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1': XOR Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>30:28</td>
<td></td>
<td><strong>VMCRunit clock gate disable</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 111b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VMCR unit Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>27:25</td>
<td></td>
<td><strong>SMCRunit clock gate disable</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 111b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SMCR unit Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>24:22</td>
<td></td>
<td><strong>MCRunit clock gate disable</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 111b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MCR unit Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>Register</td>
<td>Description</td>
<td>Access</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------------------------------------------</td>
<td>--------</td>
</tr>
<tr>
<td><strong>RCGCTL2</strong></td>
<td>RAM Clock Gating Control 2</td>
<td>R/W</td>
</tr>
<tr>
<td><strong>MVUunit</strong></td>
<td>RAM clock gate disable</td>
<td>R/W</td>
</tr>
<tr>
<td><strong>WAVM unit</strong></td>
<td>RAM clock gate disable</td>
<td>R/W</td>
</tr>
<tr>
<td><strong>WHME unit</strong></td>
<td>RAM clock gate disable bit</td>
<td>R/W</td>
</tr>
<tr>
<td><strong>WIME unit</strong></td>
<td>RAM clock gate disable</td>
<td>R/W</td>
</tr>
<tr>
<td><strong>WMPC unit</strong></td>
<td>RAM clock gating disable</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>RCGCTL2 - RAM Clock Gating Control 2</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>-------------------------------------</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td><strong>SDEunit RAM clock gate disable</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Access:</strong> R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SDE unit RAM Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td><strong>VSHM unit clock gate disable</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Access:</strong> R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td>VSHM unit RAM Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td><strong>DAPRTS unit RAM clock gate disable</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Access:</strong> R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DAPRTS unit RAM Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1': Clock Gating Disabled.</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td><strong>GS unit RAM clock gate disable</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Access:</strong> R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td>GS unit RAM Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td><strong>Reserved</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Access:</strong> R/W</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td><strong>GAMTunit RAM clock gate disable bit</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Access:</strong> R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td>GAMT unit RAM Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td><strong>VCW unit RAM clock gate disable</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Access:</strong> R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td>VCW unit RAM Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
</tbody>
</table>
## RCGCTL2 - RAM Clock Gating Control 2

<table>
<thead>
<tr>
<th></th>
<th>VEO unit RAM clock gate disable</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Access:</strong></td>
<td>R/W</td>
</tr>
<tr>
<td>VEO unit RAM Clock Gating Disable Control:</td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>IMEunit RAM clock gate disable</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Access:</strong></td>
<td>R/W</td>
</tr>
<tr>
<td>IMEunit RAM Clock Gating Disable Control:</td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>CREunit RAM clock gate disable</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Access:</strong></td>
<td>R/W</td>
</tr>
<tr>
<td>CREunit RAM Clock Gating Disable Control:</td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>RSunit RAM clock gate disable</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Access:</strong></td>
<td>R/W</td>
</tr>
<tr>
<td>RSunit RAM Clock Gating Disable Control:</td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>MSCunit RAM Clock Gating Disable</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Access:</strong></td>
<td>R/W</td>
</tr>
<tr>
<td>MSCunit RAM Clock Gating Disable Control:</td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>VMXunit RAM Clock Gating Disable</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Access:</strong></td>
<td>R/W</td>
</tr>
<tr>
<td>VMXunit RAM Clock Gating Disable Control:</td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>2</td>
<td><strong>GAunit RAM Clock Gating Disable for all EUs</strong></td>
</tr>
<tr>
<td>---</td>
<td>---------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td>GAunit RAM Clock Gating Disable Control For all EUs in each Row:</td>
</tr>
<tr>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1</th>
<th><strong>VSunit RAM Clock Gating Disable</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td>VSunit RAM Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>0</th>
<th><strong>HSunit RAM Clock Gating Disable</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td>HSunit RAM Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>
## RAM Clock Gating Control 3

<table>
<thead>
<tr>
<th><strong>Register Space:</strong></th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Project:</strong></td>
<td>CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong></td>
<td>PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong></td>
<td>0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong></td>
<td>32</td>
</tr>
<tr>
<td><strong>Address:</strong></td>
<td>09434h</td>
</tr>
</tbody>
</table>

### DWord 0: 31:13

- **RSVD**
  - **Access:** RO
  - **Description:** Reserved

### DWord 12

- **Reserved**
  - **Access:**

### DWord 11

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

### DWord 10

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

### DWord 9

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

### DWord 8

- **cp_ramcgdis_hpr**
  - **Access:** R/W
  - **Description:** HPR unit Clock Gating Disable (cp_ramcgdis_hpr)
  - HPR unit Clock Gating Disable Control
  - '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)
### RCGCTL3 - RAM Clock Gating Control 3

<table>
<thead>
<tr>
<th>Address</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x4324</td>
<td>R/W</td>
<td>cp_ramcgdis_hmc  HMC unit Clock Gating Disable (cp_ramcgdis_hmc) HMC unit Clock Gating Disable Control '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>0x4325</td>
<td>R/W</td>
<td>cp_ramcgdis_hlf  HLF unit Clock Gating Disable (cp_ramcgdis_hlf) HLF unit Clock Gating Disable Control '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>0x4326</td>
<td>R/W</td>
<td>cp_ramcgdis_hmx  HMX unit Clock Gating Disable (cp_ramcgdis_hmx) HMX unit Clock Gating Disable Control '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>0x4327</td>
<td>R/W</td>
<td>cp_ramcgdis_vmm  VMM unit Clock Gating Disable (cp_ramcgdis_vmm) VMM unit Clock Gating Disable Control '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>0x4328</td>
<td>R/W</td>
<td>cp_ramcgdis_mpd  MPD unit Clock Gating Disable (cp_ramcgdis_mpd) MPD unit Clock Gating Disable Control '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>
### RCGCTRL3 - RAM Clock Gating Control 3

<table>
<thead>
<tr>
<th>Offset</th>
<th>Register Description</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td><strong>cp_ramcgdis_mbd</strong></td>
<td>R/W</td>
<td>MBD unit Clock Gating Disable (cp_ramcgdis_mbd)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>MBD unit Clock Gating Disable Control</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>0001</td>
<td><strong>cp_ramcgdis_mmx</strong></td>
<td>R/W</td>
<td>MMX unit Clock Gating Disable (cp_ramcgdis_mmx)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>MMX unit Clock Gating Disable Control</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>0002</td>
<td><strong>cp_ramcgdis_vmpc</strong></td>
<td>R/W</td>
<td>VMPC unit Clock Gating Disable (cp_ramcgdis_vmpc)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>VMPC unit Clock Gating Disable Control</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>
**RC6 Wake Rate Limit**

<table>
<thead>
<tr>
<th><strong>RCXWRL - RC6 Wake Rate Limit</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
</tbody>
</table>

**Address:** 0A09Ch-0A09Fh

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td><strong>RC6WRL</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RC Promotion Time Modulated by wake rate limits when using EI method:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Even though the RC6 promotion time is met, if the wake limit is exceeded, no promotion pmcr_rc6_wake_rate_limit[15:0]</td>
</tr>
<tr>
<td>15:0</td>
<td>Reserved</td>
<td><strong>Access:</strong> RO</td>
</tr>
</tbody>
</table>
# RCC LRA 0

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td>RCC LRA1 Max</td>
</tr>
<tr>
<td></td>
<td>Default Value:</td>
<td>01111111b</td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>Maximum value of programmable LRA1.</td>
<td></td>
</tr>
<tr>
<td>23:16</td>
<td>RCC LRA0 Min</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default Value:</td>
<td>01000000b</td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>Minimum value of programmable LRA1.</td>
<td></td>
</tr>
<tr>
<td>15:8</td>
<td>RCC LRA0 Max</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default Value:</td>
<td>00111111b</td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>Maximum value of programmable LRA0.</td>
<td></td>
</tr>
<tr>
<td>7:0</td>
<td>RCC LRA0 Min</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default Value:</td>
<td>00000000b</td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>Minimum value of programmable LRA0.</td>
<td></td>
</tr>
</tbody>
</table>
### RCC LRA 1

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000001</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>04A44h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:2</td>
<td><strong>Reserved</strong></td>
<td>00000000000000000000000000000000b</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td><strong>MSC LRA</strong></td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Which LRA should MSC use.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td><strong>RCC LRA</strong></td>
<td>1b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Which LRA should RCC use.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### RCC Virtual page Address Registers

<table>
<thead>
<tr>
<th>RCCTLB_VA - RCC Virtual page Address Registers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: RenderCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Access: RO</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Trusted Type: 1</td>
</tr>
<tr>
<td>Address: 04A00h-04A03h</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:12</td>
<td>Address</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: All</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: GraphicsAddress[31:12]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Page virtual address.</td>
</tr>
<tr>
<td>0</td>
<td>11:0</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
</tbody>
</table>
## RC Evaluation Interval

### RCI - RC Evaluation Interval

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0A0A8h-0A0ABh</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>23:0</td>
<td></td>
<td><strong>Render Standby Evaluation Interval</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 = 0 usec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 = 1.28 usec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 = 2.56 usec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 = 3.84 usec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FF FFFF = 21.474 sec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>pmcr_rc_ei[23:0]</td>
</tr>
</tbody>
</table>
## RC Idle Hysteresis

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td>23:0</td>
<td></td>
<td><strong>RC Idle Hysteresis Detection</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
</tbody>
</table>

Idle intervals must be longer than this value to be considered idle.  
0 = 0 usec means disabled  
1 = 1.28 usec  
2 = 2.56 usec  
3 = 3.84 usec  
FF FFFF = 21.474 sec  
This must not be set to more than 5ms to prevent the PCU from timing out on an S state entry
# RCS_PREEMPTION_HINT

<table>
<thead>
<tr>
<th>RCS_PREEMPTION_HINT - RCS_PREEMPTION_HINT</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> RenderCS</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 024BCh</td>
</tr>
</tbody>
</table>

### Description

This register contains the Head pointer offset in to the Ring Buffer or the Dword aligned Graphics address in to the Batch Buffer corresponding to either MI_ARB_CHECK called Preemption Hint Address. When Preemption Hint Address is enabled, RCS will honor UHPRTR only on parsing MI_ARB_CHK at Preemption Hint Address.

This register contains the Head pointer offset into the Ring Buffer or the Dword aligned Graphics address into the Batch Buffer corresponding to the below listed commands in exeqlist mode of operation:

- MI_ARB_CHECK
- MI_WAIT_FOR_EVENT
- MI_SEMAPHORE_WAIT
- 3D_PRIMITIVE
- GPGPU_WALKER
- MEDIA_STATE_FLUSH
- PIPE_CONTROL (Only in GPGPU mode of pipeline selection)
- MI_ATOMIC (Post Sync Operation set in GPGPU mode of pipeline selection)
- MI_SEMAPHORE_SIGNAL (Post Sync Operation set in GPGPU mode of pipeline selection)

### Programming Notes

**Programming Restriction:**
This register should NEVER be programmed in functional mode, this should be used only in validation mode to achieve deterministic behavior of UHPRTR being sampled by a given MI_ARB_CHECK in command stream. Programmer has to ensure that RCS Preemption Hint register gets programmed before UHPRTR is programmed and well before RCS crosses the corresponding execution point. Preemption hint for both RingBuffer and Batch Buffer can’t be enabled simultaneously.

User must ensure the Preempted Hint Address programmed matches either Ring Head Offset or Batch Buffer Graphics Virtual Address and not both of them.
User must also ensure the Preempted Hint Address[19:0] programmed matches either Ring Head Offset[19:0] or Batch Buffer Graphics Virtual Address[19:0] and not both of them.
### RCS_PREEMPTION_HINT - RCS_PREEMPTION_HINT

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:2</td>
<td><strong>Preempted Hint Address</strong></td>
<td>CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: CHV, BSW</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: U30</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>This field contains the Head offset in to the Ring Buffer when Preemption Hint is set to Ring Buffer and Dword aligned Graphics Address in to the batch buffer when Preemption Hint is set to Batch Buffer.</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td><strong>Batch Buffer Preemption Hint</strong></td>
<td>CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: CHV, BSW</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: Enabled</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Value</strong></td>
<td><strong>Name</strong></td>
</tr>
<tr>
<td>0h</td>
<td></td>
<td>Disabled</td>
<td>Preemption hint is disabled in batch buffer.</td>
</tr>
<tr>
<td>1h</td>
<td></td>
<td>Enabled</td>
<td>Preemption hint is enabled in ring buffer and preemption hint address corresponds to the instruction in Batch Buffer.</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td><strong>Ring Preemption Hint</strong></td>
<td>CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: CHV, BSW</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: Enable</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Value</strong></td>
<td><strong>Name</strong></td>
</tr>
<tr>
<td>0h</td>
<td></td>
<td>Disable</td>
<td>Preemption hint is disabled in ring buffer.</td>
</tr>
<tr>
<td>1h</td>
<td></td>
<td>Enabled</td>
<td>Preemption hint is enabled in ring buffer and preemption hint address corresponds to the instruction in Ring Buffer.</td>
</tr>
</tbody>
</table>
RCS_PREEMPTION_HINT_UDW

<table>
<thead>
<tr>
<th>RCS_PREEMPTION_HINT_UDW - RCS_PREEMPTION_HINT_UDW</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: RenderCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Access: R/W</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 024C8h</td>
</tr>
</tbody>
</table>

This register contains the 4GB aligned base address of gfx 4GB virtual address space within the host’s 64-bit virtual address space of the Batch Buffer corresponding to MI_ARB_CHECK command called Preemption Hint Address.

Programming Notes

Programming Restriction:
This register should NEVER be programmed in functional mode, this should be used only in validation mode to achieve deterministic behavior of UHPTR being sampled by a given MI_ARB_CHK in command stream.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td>Format: MBZ</td>
<td></td>
</tr>
<tr>
<td>15:0</td>
<td>Preempted Hint Address Upper DWORD</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Format: GraphicsAddress[47:32]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>This field contains the 4GB aligned base address of gfx 4GB virtual address space within the host’s 64-bit virtual address space of the batch buffer when Preemption Hint is set to Batch Buffer. This field is not valid when Preemption Hint is set to Ring Buffer.</td>
<td></td>
</tr>
</tbody>
</table>
# RCS Batch Buffer State Register

<table>
<thead>
<tr>
<th><strong>RCS_BB_STATE - RCS Batch Buffer State Register</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: RenderCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000 CHV, BSW</td>
</tr>
<tr>
<td>Access: RO</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 02110h</td>
</tr>
</tbody>
</table>

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

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:9</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td><strong>Resource Streamer Enable</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: U1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>When this bit is set, the Resource Streamer will execute the batch buffer. When this bit is clear the Resource Streamer will not execute the batch buffer.</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td><strong>Address Space Indicator</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note: 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.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Value</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1h</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td>3:0</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
</tbody>
</table>

---

*Doc Ref # IHD-OS-CHV-BSW-Vol 2c-10.15*
RCS Context Preemption Hint

<table>
<thead>
<tr>
<th>RCS_CTXID_PREEMPTION_HINT - RCS Context Preemption Hint</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: RenderCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Access: R/W</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 024CCh</td>
</tr>
<tr>
<td>Address: 124CCh-124CFh</td>
</tr>
<tr>
<td>Name: Context ID Preemption Hint</td>
</tr>
<tr>
<td>ShortName: RCS_CTXID_PREEMPTION_HINT_VCSUNIT0</td>
</tr>
<tr>
<td>Address: 1A4CCh-1A4CFh</td>
</tr>
<tr>
<td>Name: Context ID Preemption Hint</td>
</tr>
<tr>
<td>ShortName: RCS_CTXID_PREEMPTION_HINT_VECSUNIT</td>
</tr>
<tr>
<td>Address: 1C4CCh-1C4CFh</td>
</tr>
<tr>
<td>Name: Context ID Preemption Hint</td>
</tr>
<tr>
<td>ShortName: RCS_CTXID_PREEMPTION_HINT_VCSUNIT1</td>
</tr>
<tr>
<td>Address: 224CCh-224CFh</td>
</tr>
<tr>
<td>Name: Context ID Preemption Hint</td>
</tr>
<tr>
<td>ShortName: RCS_CTXID_PREEMPTION_HINT_BCSUNIT</td>
</tr>
</tbody>
</table>

This register contains the Context ID of a context in exectlist mode of operation. In exectlist mode of operation RCS_PREEMPTION_HINT and RS_PREEMPTION_HINT registers are looked at by Render Command Streamer and Resource Streamer only on executing a context having Context ID that matches with the contents of this register. This register contents are valid and looked at only in exectlist mode of operation.

**Programming Restriction:**
This register should NEVER be programmed in functional mode, this must be used only in validation mode to achieve deterministic preemption behavior in exectlist mode of operation.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>Context ID Preemption Hint</td>
</tr>
</tbody>
</table>

Format: U32

If 0 this field has no effect. If nonzero it indicates the only context ID that can be preempted when exectlists are enabled. A preemption attempt when the context ID of the currently executing ring context does not match this field will be ignored.
## RC Wake Counter

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space</td>
<td>MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits)</td>
<td>32</td>
</tr>
<tr>
<td>Address</td>
<td>0A0A4h-0A0A7h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td><strong>Wake Counter Render</strong></td>
<td>RO</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Incremented for each wake event, wraps around wake_counter[15:0]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>15:0</td>
<td><strong>Wake Counter Media</strong></td>
<td>RO</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Incremented for each wake event, wraps around wake_counter[15:0]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
RCZ Virtual Page Address Registers

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:12</td>
<td>Address</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: GraphicsAddress[31:12]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Page virtual address.</td>
</tr>
<tr>
<td>11:0</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
</tbody>
</table>
Ready Bit Vector 0 for TLBPEND registers

<table>
<thead>
<tr>
<th>TLBPEND_RDY0 - Ready Bit Vector 0 for TLBPEND registers</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> RenderCS</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Trusted Type:</strong> 1</td>
</tr>
<tr>
<td><strong>Address:</strong> 04708h-0470Bh</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>Ready bits per entry</td>
</tr>
</tbody>
</table>
Ready Bit Vector 1 for TLBPEND registers

<table>
<thead>
<tr>
<th>TLBPEND_RDY1 - Ready Bit Vector 1 for TLBPEND registers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: RenderCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Access: R/W</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Trusted Type: 1</td>
</tr>
<tr>
<td>Address: 0470Ch-0470Fh</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Ready bits per entry</strong></td>
</tr>
</tbody>
</table>
## Render C State Control 1

### RCCTL1 - Render C State Control 1

- **Register Space:** MMIO: 0/2/0
- **Project:** CHV, BSW
- **Source:** PRM
- **Default Value:** 0x00000000
- **Size (in bits):** 32

**Address:** 0A090h-0A093h

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:29</td>
<td>Reserved</td>
<td>RO</td>
</tr>
<tr>
<td>28</td>
<td></td>
<td><strong>TO - HW RC Promotion (i.e., Depth) Selection</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0: Timeout method disabled</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1: TO method enabled</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>pmcr_to_enable</td>
<td>R/W</td>
</tr>
<tr>
<td>27</td>
<td></td>
<td><strong>EI - HW RC Promotion (i.e., Depth) Selection</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0: EI disabled</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1: Evaluation Interval (EI) method enabled</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>pmcr_ei_enable</td>
<td>R/W</td>
</tr>
<tr>
<td>26:25</td>
<td></td>
<td>Reserved</td>
<td>RO</td>
</tr>
<tr>
<td>24</td>
<td></td>
<td>Reserved</td>
<td>R/W</td>
</tr>
<tr>
<td>23</td>
<td></td>
<td>Reserved</td>
<td>R/W</td>
</tr>
<tr>
<td>22:0</td>
<td></td>
<td>Reserved</td>
<td>RO</td>
</tr>
</tbody>
</table>
## Render forcewake acknowledge

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
</tbody>
</table>

**Address:** 1300B4h

This register contains the per thread force wake acknowledge bits for the Render power well.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td>RESERVED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved.</td>
</tr>
<tr>
<td>15:0</td>
<td>FWAKERENDERACK</td>
<td>Force Wake Render request bits. Driver must poll on the corresponding bit to confirm that the well has woken. For example, if 13_00B0[0] is written to a ‘1’ (along with 13_00B0[16]=’1’), then bit0 of this register indicates when the force wake request has been completed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
</tbody>
</table>
Render forcewake request

**RENFW_REQ - Render forcewake request**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>1300B0h</td>
</tr>
</tbody>
</table>

This register contains per thread force wake request bits for the Render power Well. The upper 16 bits act as masks for the lower 16 bits. Bit 31 masks bit 15 and bit 16 masks bit 0.

1. Driver writes to GPM force wake request bit. (VV will have a Render(13_0080(15:0)) and a Media (13_00B8(15:0)) bits.)
2. The GPM responds by writing (via PLINK) to 1300B4[15:0] or 1300BC[15:0] register.
3. Driver polls (1300B4[15:0] and/or 1300BC[15:0])status until 1... indicating that that well has completed wake sequence.

Since the registers are per thread, only the specific bit that was forced should be checked for status.

### DWord Bit Description

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td>FWAKERENDERREQMSK</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Mask bits for lower 16 bits to avoid a read modify/write. If '0', the corresponding bit in [15:0] is not changed. If '1', the corresponding bit in [15:0] is changed to the value in [15:0]</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>FWAKERENDERREQ15</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Force Wake Render request 15.</td>
</tr>
<tr>
<td>14</td>
<td></td>
<td>FWAKERENDERREQ14</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Force Wake Render request 14.</td>
</tr>
<tr>
<td>13</td>
<td></td>
<td>FWAKERENDERREQ13</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Force Wake Render request 13.</td>
</tr>
</tbody>
</table>
### RENFW_REQ - Render forcewake request

<table>
<thead>
<tr>
<th></th>
<th>FWAKERENDERREQ</th>
<th>Default Value</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>FWAKERENDERREQ12</td>
<td>0b</td>
<td>R/W</td>
<td>Force Wake Render request 12.</td>
</tr>
<tr>
<td>11</td>
<td>FWAKERENDERREQ11</td>
<td>0b</td>
<td>R/W</td>
<td>Force Wake Render request 11.</td>
</tr>
<tr>
<td>10</td>
<td>FWAKERENDERREQ10</td>
<td>0b</td>
<td>R/W</td>
<td>Force Wake Render request 10.</td>
</tr>
<tr>
<td>9</td>
<td>FWAKERENDERREQ9</td>
<td>0b</td>
<td>R/W</td>
<td>Force Wake Render request 9.</td>
</tr>
<tr>
<td>8</td>
<td>FWAKERENDERREQ8</td>
<td>0b</td>
<td>R/W</td>
<td>Force Wake Render request 8.</td>
</tr>
<tr>
<td>7</td>
<td>FWAKERENDERREQ7</td>
<td>0b</td>
<td>R/W</td>
<td>Force Wake Render request 7.</td>
</tr>
<tr>
<td>6</td>
<td>FWAKERENDERREQ6</td>
<td>0b</td>
<td>R/W</td>
<td>Force Wake Render request 6.</td>
</tr>
<tr>
<td>5</td>
<td>FWAKERENDERREQ5</td>
<td>0b</td>
<td>R/W</td>
<td>Force Wake Render request 5.</td>
</tr>
</tbody>
</table>
## RENFW_REQ - Render forcewake request

<table>
<thead>
<tr>
<th>Bit</th>
<th>Field</th>
<th>Default Value</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>FWAKERENDERREQ4</td>
<td>0b</td>
<td>R/W</td>
<td>Force Wake Render request 4.</td>
</tr>
<tr>
<td>3</td>
<td>FWAKERENDERREQ3</td>
<td>0b</td>
<td>R/W</td>
<td>Force Wake Render request 3.</td>
</tr>
<tr>
<td>2</td>
<td>FWAKERENDERREQ2</td>
<td>0b</td>
<td>R/W</td>
<td>Force Wake Render request 2.</td>
</tr>
<tr>
<td>1</td>
<td>FWAKERENDERREQ1</td>
<td>0b</td>
<td>R/W</td>
<td>Force Wake Render request 1.</td>
</tr>
<tr>
<td>0</td>
<td>FWAKERENDERREQ0</td>
<td>0b</td>
<td>R/W</td>
<td>Force Wake Render request 0.</td>
</tr>
</tbody>
</table>
## Render Geyserville Mode Control 1

### RPMODECTL1 - Render Geyserville Mode Control 1

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0A024h-0A027h</td>
</tr>
</tbody>
</table>

### DWord 0

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>31:12</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

### DWord 11

#### RP Video Turbo Enable for Media Engine

Access:

Refer to bits A024[10:9] for different Media turbo scenarios.
Refer to the table in the description for bits A024[10:9].

### DWord 10:9

#### RP Software Mode Control

Access:

11 - Media_active (8000[6]), Media_done and RPVTENMEDIA(A024[11]) are don't care. Media turbo freq is selected.
10 - Media_active (8000[6]), Media_done and RPVTENMEDIA(A024[11]) are don't care. Normal freq(Hard coded)
01 - Based on the status either the media-turbo or normal frequency will be selected.
00 - Media_active (8000[6]), Media_done and RPVTENMEDIA(A024[11]) are don't care. Current frequency is maintained.

<table>
<thead>
<tr>
<th>pmcr_rpsw_ctl_mode[1]</th>
<th>pmcr_rpsw_ctl_mode[0]</th>
<th>pmmr_media_active</th>
<th>pmcr_rpsw_vten_vcs</th>
<th>pmmr_vcs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Command Reference: Registers

## RPMODECTL1 - Render Geyserville Mode Control

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>

### CHV, BSW description

<table>
<thead>
<tr>
<th>pmcr_rpsw_ctl_mode[1]</th>
<th>pmcr_rpsw_ctl_mode[0]</th>
<th>pmmr_media_active</th>
<th>pmcr_rpsw_vten_vcs</th>
<th>pmmr_vcs_done</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>

### 8 Mask Bits for Graphics Busyness

**Access:**

<table>
<thead>
<tr>
<th>Mask Bits for Graphics Busyness</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mask Bits for Graphics Busyness (MBGB) and enables MC0 counter.</td>
<td></td>
</tr>
<tr>
<td>0: MFX busyness is not counted as part of gfx busyness and media MC0 counter forced to a zero.</td>
<td></td>
</tr>
<tr>
<td>1: MFX busyness is counted as part of gfx busyness and media MC0 counter is allowed to count.</td>
<td></td>
</tr>
<tr>
<td>Both VCS and VECS is counted towards media busyness. BIOS/driver should always set this bit to a one.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>pmcr_media_mask</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>7 Render Geyserville HW Controlled Idle Mode Enable</td>
<td></td>
</tr>
</tbody>
</table>

**Access:**

<table>
<thead>
<tr>
<th>Render Geyserville HW Controlled Idle Mode Enable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 - Disables Render Geyserville (RP) function (default).</td>
<td></td>
</tr>
<tr>
<td>1 - Turns on the Render Geyserville (RP) function and RP counters are enabled (program enable after counter writing)</td>
<td></td>
</tr>
</tbody>
</table>
### RPMODECTL1 - Render Geyserville Mode Control

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td>5:3</td>
<td><strong>Frequency Increase Utilization Metric Selection</strong></td>
</tr>
<tr>
<td>2:0</td>
<td><strong>Frequency Decrease Utilization Metric Selection</strong></td>
</tr>
</tbody>
</table>

#### Frequency Increase Utilization Metric Selection

Access: 

The selection of a metric below indicates which one is to be used in making decision. All the metrics should be running and updating the status counters regardless of this setting.

- More than Busy Max Continuous (BMXC) time must be reached for a frequency increase
- More than Busy Max Average (BMXA) at end of Evaluation Interval must be reached for a frequency increase
- Less than Idle Min Continuous (IMNC) time must be reached for a frequency increase

BMXC BMXA IMNC

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0</td>
<td>no metric enabled</td>
</tr>
<tr>
<td>0 0 1</td>
<td>IMNC metric enabled (0 % val)</td>
</tr>
<tr>
<td>0 1 0</td>
<td>BMXA metric enabled (90 % val)</td>
</tr>
<tr>
<td>0 1 1</td>
<td>reserved</td>
</tr>
<tr>
<td>1 0 0</td>
<td>BMXC metric enabled (10 % val)</td>
</tr>
<tr>
<td>1 0 1</td>
<td>reserved</td>
</tr>
<tr>
<td>1 1 0</td>
<td>reserved</td>
</tr>
<tr>
<td>1 1 1</td>
<td>reserved</td>
</tr>
</tbody>
</table>

pmcr_freq_inc_utimet[2:0]

#### Frequency Decrease Utilization Metric Selection

Access: 

The selection of a metric below indicates which one is to be used in making decision. All the metrics should be running and updating the status counters regardless of this setting.

- Less than Busy Min Continuous must be reached for a frequency decrease
- Less than Busy Min Average at end of Evaluation Interval must be reached for a frequency decrease
- More than Idle Max Continuous must be reached for a frequency decrease

BMNC BMNA IMXC

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0</td>
<td>no metric enabled</td>
</tr>
<tr>
<td>0 0 1</td>
<td>IMXC metric enabled (0 % val)</td>
</tr>
<tr>
<td>0 1 0</td>
<td>BMNA metric enabled (100 % val)</td>
</tr>
<tr>
<td>0 1 1</td>
<td>reserved</td>
</tr>
<tr>
<td>1 0 0</td>
<td>BMNC metric enabled (0% val)</td>
</tr>
<tr>
<td>1 0 1</td>
<td>reserved</td>
</tr>
<tr>
<td>1 1 0</td>
<td>reserved</td>
</tr>
<tr>
<td>1 1 1</td>
<td>reserved</td>
</tr>
</tbody>
</table>

pmcr_freq_dec_utimet[2:0]
# MI_MODE - Render Mode Register for Software Interface

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>RenderCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0209Ch</td>
</tr>
<tr>
<td>Address:</td>
<td>1209Ch-1209Fh</td>
</tr>
<tr>
<td>Name:</td>
<td>Mode Register for Software Interface</td>
</tr>
<tr>
<td>ShortName:</td>
<td>MI_MODE_VCSUNIT0</td>
</tr>
<tr>
<td>Address:</td>
<td>1A09Ch-1A09Fh</td>
</tr>
<tr>
<td>Name:</td>
<td>Mode Register for Software Interface</td>
</tr>
<tr>
<td>ShortName:</td>
<td>MI_MODE_VECSUNIT</td>
</tr>
<tr>
<td>Address:</td>
<td>1C09Ch-1C09Fh</td>
</tr>
<tr>
<td>Name:</td>
<td>Mode Register for Software Interface</td>
</tr>
<tr>
<td>ShortName:</td>
<td>MI_MODE_VCSUNIT1</td>
</tr>
<tr>
<td>Address:</td>
<td>2209Ch-2209Fh</td>
</tr>
<tr>
<td>Name:</td>
<td>Mode Register for Software Interface</td>
</tr>
<tr>
<td>ShortName:</td>
<td>MI_MODE_BCSUNIT</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td><strong>Masks</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: Mask[15:0]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A 1 in a bit in this field allows the modification of the corresponding bit in Bits 15:0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>15</td>
<td><strong>Suspend Flush</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: U1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>No Delay</td>
<td>HW will not delay flush, this bit will get cleared by MI_SUSPEND_FLUSH as well</td>
</tr>
<tr>
<td></td>
<td>[Default]</td>
<td></td>
</tr>
<tr>
<td>1h</td>
<td>Delay Flush</td>
<td>Suspend flush is active</td>
</tr>
</tbody>
</table>
### MI_MODE - Render Mode Register for Software Interface

**Programming Notes**

This should only be written to from the ring using MI_SUSPEND_FLUSH. It is considered undefined if written by software through MMIO.

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>Performance mode enabled</td>
<td>The stall of the flip event is in the windower</td>
</tr>
<tr>
<td></td>
<td>[Default]</td>
<td></td>
</tr>
<tr>
<td>1h</td>
<td>Performance mode disabled</td>
<td>The stall of the flip event is in the command stream</td>
</tr>
</tbody>
</table>

**Programming Notes**

This bit must be set to ‘1’ on all projects disabling Async Flip Performance mode.

When Async Flip Performance mode is enabled stall is in the Windower allowing the commands following the MI_WAIT_FOR_EVENT to be parsed by command streamer, this breaks the usage model of controlling the display message generation in display engine using MI_LOAD_REGISTER_IMMEDIATE commands from ring buffer.

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>run fast restore [Default]</td>
<td>No NonPipelined SV flush.</td>
</tr>
<tr>
<td>1h</td>
<td>run slow legacy restore</td>
<td>With NonPipelined SV flush.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Reserved</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Invalidate UHPTR enable</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Atomic Read Return for MI_COPY_MEM_MEM</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Project: CHV, BSW</th>
<th>Form: U1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>Name</td>
</tr>
<tr>
<td>0h</td>
<td>Disable</td>
</tr>
<tr>
<td></td>
<td>[Default]</td>
</tr>
<tr>
<td>1h</td>
<td>Enable</td>
</tr>
</tbody>
</table>
### MI_MODE - Render Mode Register for Software Interface

#### 9 Rings Idle

**Format:** U1

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>Not Idle [Default]</td>
<td>Parser not Idle or Ring Arbiter not Idle.</td>
</tr>
<tr>
<td>1h</td>
<td>Idle</td>
<td>Parser Idle and Ring Arbiter Idle.</td>
</tr>
</tbody>
</table>

**Programming Notes**

Writes to this bit are not allowed.

#### 8 Stop Rings

**Format:** U1

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>[Default]</td>
<td>Normal Operation.</td>
</tr>
<tr>
<td>1h</td>
<td></td>
<td>Parser is turned off and Ring arbitration is turned off.</td>
</tr>
</tbody>
</table>

**Programming Notes**

Software must set this bit to force the Rings and Command Parser to Idle. Software must read a 1 in the Ring Idle bit after setting this bit to ensure that the hardware is idle.

Software must clear this bit for Rings to resume normal operation.

Due to known HW issue when Stop Rings occur during execution of a batch buffer, memory access type of the batch buffer is reset and hence on resuming the memory access type can be inconsistent with the desired memory access type. SW must not set/reset Stop Rings to achieve stall and resume function in command streamer execution, however Stop Rings can be used by SW before resetting the engine.

#### 7 Reserved

**Format:** MBZ

#### 6 Vertex Shader Timer Dispatch Enable

**Project:** CHV, BSW

**Format:** Enable

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>Disable [Default]</td>
<td>Disable the timer for dispatch of single vertices from the vertex shader. Vertex shader will try to collect 2 vertices before a dispatch</td>
</tr>
<tr>
<td>1h</td>
<td>Enable</td>
<td>Enable the timer for dispatch of single vertices. Dispatch a single vertex shader thread after the timer expires.</td>
</tr>
</tbody>
</table>

#### 5 Reserved

**Format:** MBZ
### MI_MODE - Render Mode Register for Software Interface

#### 4:1 Predicate Enable

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>Predicate Disable</td>
<td>Predication is Disabled and RCS will process commands as usual.</td>
</tr>
<tr>
<td>1h</td>
<td>Predicate on Result2 clear</td>
<td>Following Commands will be NOOPED by RCS only if the MI_PREDICATE_RESULT_2 is clear.</td>
</tr>
<tr>
<td>2h</td>
<td>Predicate on Result2 set</td>
<td>Following Commands will be NOOPED by RCS only if the MI_PREDICATE_RESULT_2 is set.</td>
</tr>
<tr>
<td>3h</td>
<td>Predicate on Result clear</td>
<td>Following Commands will be NOOPED by RCS only if the MI_PREDICATE_RESULT is clear.</td>
</tr>
<tr>
<td>4h</td>
<td>Predicate on Result set</td>
<td>Following Commands will be NOOPED by RCS only if the MI_PREDICATE_RESULT is set.</td>
</tr>
<tr>
<td>5h</td>
<td>Predicate when two or more slices enabled</td>
<td>Following Commands will be NOOPED by RCS only when one slice is enabled, NOOPED when more than one slice is enabled.</td>
</tr>
<tr>
<td>6h</td>
<td>Predicate when one or three slices enabled</td>
<td>Following Commands will be Executed by RCS only when two slices are enabled, NOOPED when one or three slices are enabled.</td>
</tr>
<tr>
<td>7h</td>
<td>Predicate when one or two slices enabled</td>
<td>Following Commands will be Executed by RCS only when all the three slices are enabled, NOOPED when less than three slices are enabled.</td>
</tr>
<tr>
<td>8h-Eh</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>Fh</td>
<td>Predicate Always</td>
<td>Following Commands will be NOOPED by RCS unconditionally.</td>
</tr>
</tbody>
</table>

#### Programming Notes

SW must use MI_SET_PREDICATE instead of MMIO access.

#### 0 Mask IIR disable

<table>
<thead>
<tr>
<th>Format</th>
<th>Disable</th>
</tr>
</thead>
</table>

Mask IIR disable. Nominally the Interrupt controller masks interrupts in the IIR register if an interrupt acknowledge from the 3gio interface is pending. Setting this bit to a 1 allows interrupts to be visible to the interrupt controller while an interrupt acknowledge is pending.
Render Performance Status 1

### RPSTAT1 - Render Performance Status 1

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td><strong>Reserved</strong></td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access:</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td></td>
<td><strong>Last Requested Video Turbo Mode</strong></td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Last Requested Video Turbo Mode (CRTM):</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 = Most recent request was a normal request (from RPNSWREQ)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 = Most recent request was a Video Turbo request (from RPVSWREQ)</td>
<td></td>
</tr>
<tr>
<td>14:8</td>
<td></td>
<td><strong>Reserved</strong></td>
<td>RO</td>
</tr>
<tr>
<td>7:0</td>
<td></td>
<td><strong>Reserved</strong></td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access:</td>
<td></td>
</tr>
</tbody>
</table>

This register reflects real-time values and thus will not have a pre-determined default value out of reset.
## Render Performance Status Register

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
</table>
| 0     | 31:20 | **Reserved**  
Access: RO |
| 19:18 |  Current Actual Gear Ratio  
Access: RO |
| 17:16 | Previous Actual Gear Ratio  
Access: RO |
| 15    | Last Requested Video Turbo Mode  
Access: RO |
| 14    | Reserved  
Access: RO |
| 13:7  | Current Actual GFX Freq  
Access: RO |
| 6:0   | Previous Actual GFX Freq  
Access: RO |

This register reflects real-time values and thus does not have a pre-determined default value out of reset.

- **Reserved**: Accesses are read-only (RO).
- **Current Actual Gear Ratio**: Previous Actual Gear Ratio (PAGR).
- **Last Requested Video Turbo Mode**: Last Requested Video Turbo Mode (CRTM):  
  - 0 = Most recent request was a normal request (from RPNSWREQ).
  - 1 = Most recent request was a Video Turbo request (from RPVSWREQ).
- **Current Actual GFX Freq**: This is the MLC ratio that the core is actually running.
- **Previous Actual GFX Freq**: This is the MLC ratio that the core was actually running before the current actual GFX frequency.
### Render Power Clock State Register

<table>
<thead>
<tr>
<th>R_PWR_CLK_STATE - Render Power Clock State Register</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: RenderCS</td>
</tr>
<tr>
<td>Default Value: 0x00000288 CHV, BSW</td>
</tr>
<tr>
<td>Access: R/W</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 020C8h</td>
</tr>
</tbody>
</table>

This register contains the mode selection for configuring render engine to attain desired performance and power requirements for a given context. This register is render context save/restored. This register must be initialized correctly when the context is submitted for the first time. This register is context save/restored as part of Exec-List context image in both Exec-List and Ring-Buffer mode of scheduling. This register contents are valid only when "Enable" bit [31] of the register is set.

### Programming Notes

This register must not be programmed directly through CPU MMIO cycle. Exec-List Scheduling Mode: Every context can have its own required render engine configuration by programming this register appropriately in the logical render context image in memory (LRCA) before submitting the context to the execlist submit port. This register must not be programmed using MI_LOAD_REGISTER_IMM command in ring buffer or in batch buffer, however programming "NON-SLM Indication" field through MI_LOAD_REGISTER_IMM is an exception defined below. If a need arises to change the render configuration for a context being executed in HW, Scheduler must preempt the context and update the desired render configuration in the logical render context image in memory and resubmit the context. Only "NON-SLM Indication" field in R_PWR_CLK_STATE register is allowed to be modified through MI_LOAD_REGISTER_IMM command in ring_buffer or privileged_batch_buffer. SW must modify only "NON-SLM Indication" field and must ensure to program other fields with the same value as in LRCA. SW must ensure to program PIPECONTROL flush command with CS Stall and HDC Flush prior to programming MI_LOAD_REGISTER_IMM command to modify "NON-SLM Indication" in R_PWR_CLK_STATE register. Example: //R_PWR_CLK_STATE register value in LRCA configured with two slices and NON-SLM indication reset: 0x80005_0000 //SW desires to set NON-SLM Indication filed in ring buffer MI_LOAD_REGISTER_IMM 0x20C8, 0x8005_0100 Ring Buffer Scheduling: This register must be programmed using MI_LOAD_REGISTER_IMM command in the ring buffer. When this register is being programmed to re-configure the number of slices, SW must context save the state before programming this register and restore the state after programming the register via dummy MI_SET_CONTEXT command, this will ensure the existing state is programmed to all the new slices that are powered up, in case of slice shutdown this is not required. EX: MI_SET_CONTEXT → CXTA MI_BATCH_BUFFER_START MI_BATCH_BUFFER_START MI_SET_CONTEXT → CXTB //Dummy Context to save existing render state to be restored latter. MI_LOAD_REGISTER_IMM : R_PWR_CLK_STATE (1 Slice to 3 Slices) // Slice configuration done. MI_SET_CONTEXT → CXTA // Context restore of valid state to all the slices powered up.
### R_PWR_CLK_STATE - Render Power Clock State Register

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31</td>
<td><strong>Power Clock State Enable</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Value</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1h</td>
</tr>
<tr>
<td></td>
<td>30:0</td>
<td><strong>Render Power Clock State</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Project:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format:</strong></td>
</tr>
</tbody>
</table>
## Render Power Meter Counter

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0A274h-0A277h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Access:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31</td>
<td><strong>Render Power Meter Counter Overflow</strong></td>
<td>RO</td>
<td></td>
</tr>
<tr>
<td></td>
<td>30:0</td>
<td><strong>Render PWRMTR Counter</strong></td>
<td>RO</td>
<td></td>
</tr>
</tbody>
</table>
## Render Power Meter Counter No Clear

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0A28Ch-0A28Fh</td>
</tr>
</tbody>
</table>

Formerly cleared the count and the overflow bit, but now it is just a read-only value.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31</td>
<td><strong>Render Power Meter Counter Overflow No Clear</strong></td>
<td>RO</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Formerly cleared the overflow bit, but now it is just a read-only value.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>30:0</td>
<td><strong>Render PWRMTR Counter No Clear</strong></td>
<td>RO</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Formerly cleared the count, but now is just a read-only value.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# Render Promotion Timer - RC6

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td>23:0</td>
<td></td>
<td><strong>RC6 Promote Time</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Absolute time starting from post-hyst idle.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 = 0 usec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 = 1.28 usec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 = 2.56 usec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 = 3.84 usec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FF FFFF = 21.474 sec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>pmcr_rc6_promotion_time[23:0]</td>
</tr>
</tbody>
</table>
RENDERRC0COUNTER

| Register Space: | MMIO: 0/2/0 |
| Project: | CHV, BSW |
| Source: | PRM |
| Default Value: | 0x00000000 |
| Size (in bits): | 32 |
| Address: | 138118h |

This register contains the total RC0 residency (Render powered on and clocks running) time that Render was in since boot.

SOXi Context Save/Restore : No

The 40-bit HW counter will wrap around. The only clear condition is CZ reset.

When 0x13_8104[15] is set to zero, bits 31:0 of the 40-bit counter are reported. When 0x13_8104[15] is set to one, bits 39:8 of the 40-bit counter are reported.

The units are CZ clock cycles.

It is up to SW to periodically read this register and do a difference from the last time it was read. The primary usage for this register is for power characterization.

0x13_8104[4] controls if this register should count or if it should be gated: 0 = clear, 1 = count

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>RENDERRC0TIME</td>
</tr>
</tbody>
</table>

| Default Value: | 00000000h |
| Access: | RO |

Render RC0 Residency Counter.
RENDERRC1COUNTER

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>RENDERRC1TIME</td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td>Default Value:</td>
<td>00000000h</td>
</tr>
</tbody>
</table>

This register contains the total RC1 residency (Render powered on and clock gated) time that Render was in since boot.

SOX\text{x} Context Save/Restore : No
The 40-bit HW counter will wrap around. The only clear condition is CZ reset.
When 0x13_8104[15] is set to zero, bits 31:0 of the 40-bit counter are reported. When 0x13_8104[15] is set to one, bits 39:8 of the 40-bit counter are reported.
The units are CZ clock cycles.
It is up to SW to periodically read this register and do a difference from the last time it was read. The primary usage for this register is for power characterization.

0x13_8104[2] controls if this register should count or if it should be gated: 0 = clear, 1 = count

This register contains the total RC1 residency (Render powered on and clock gated) time that Render was in since boot.

SOX\text{x} Context Save/Restore : No
The 40-bit HW counter will wrap around. The only clear condition is CZ reset.
When 0x13_8104[15] is set to zero, bits 31:0 of the 40-bit counter are reported. When 0x13_8104[15] is set to one, bits 39:8 of the 40-bit counter are reported.
The units are CZ clock cycles.
It is up to SW to periodically read this register and do a difference from the last time it was read. The primary usage for this register is for power characterization.

0x13_8104[2] controls if this register should count or if it should be gated: 0 = clear, 1 = count

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>RENDERRC1TIME</td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td>Default Value:</td>
<td>00000000h</td>
</tr>
</tbody>
</table>

Render RC1 Residency Counter.
RENDERRC6COUNTER

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>138108h</td>
</tr>
</tbody>
</table>

This register contains the total RC6 residency (Render power gated and clock gated) time that Render power well was in since boot. The counter will wrap around.

SOXi Context Save/Restore : No

The time is given in units of CZ clock cycles. The counter will reset on CZ reset going high.

This means that a warm reset will also reset this counter and it will also wrap around when it reaches max with no indication that an overflow occurred.

This register will freeze the count value (stop counting, but not reset) when pmu_gvd_renwakeack_nczfwoh=1. This register will count whenever pmu_gvd_renwakeack_nczfwoh=0.

When 0x13_8104[15] is set to zero, bits 31:0 of the 40-bit counter are reported. When 0x13_8104[15] is set to one, bits 39:8 of the 40-bit counter are reported.

The units are CZ clock cycles.

It is up to SW to periodically read this register and do a difference from the last time it was read. The primary usage for this register is for power characterization.

0x13_8104[0] controls if this register should count or if it should be gated: 0= clear, 1=count.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>RENDERRC6TIME</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Render Residency Counter.</td>
</tr>
</tbody>
</table>
## Render TLB Control Register

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:1</td>
<td>Reserved</td>
<td>0000000000000000000000000000000b</td>
<td>RO</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td>Invalidate TLBs on the corresponding Engine</td>
<td>0b</td>
<td>R/W</td>
</tr>
</tbody>
</table>

SW writes 1 to invalidate the TLBs for the associated engine and HW clears the bit when invalidation is complete. To ensure proper invalidation of the TLBs, SW has to ensure the corresponding engine’s HW pipeline is flushed and cleared from all its memory accesses. Otherwise HW cannot guarantee the proper invalidation for TLBs. This bit is self clear.
### Render Watchdog Counter

**PR_CTR - Render Watchdog Counter**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>RenderCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>RO</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
</tbody>
</table>

**Address:**
- 02190h
- 12190h-12193h
- 1A190h-1A193h
- 1C190h-1C193h
- 22190h-22193h

**Name:**
- Watchdog Counter
- Watchdog Counter
- Watchdog Counter
- Watchdog Counter

**ShortName:**
- PR_CTR_VCSUNIT0
- PR_CTR_VCSUNIT
- PR_CTR_VCSUNIT1
- PR_CTR_BCSUNIT

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Counter Value</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: U32</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This register reflects the render watchdog counter value itself. It cannot be written to.</td>
</tr>
</tbody>
</table>
## Render Watchdog Counter Threshold

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>RenderCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00150000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0217Ch</td>
</tr>
</tbody>
</table>

Address: 1217Ch-1217Fh
Name: Watchdog Counter Threshold
ShortName: PR_CTR_THRSH_VCSUNIT0

Address: 1A17Ch-1A17Fh
Name: Watchdog Counter Threshold
ShortName: PR_CTR_THRSH_VECSUNIT

Address: 1C17Ch-1C17Fh
Name: Watchdog Counter Threshold
ShortName: PR_CTR_THRSH_VCSUNIT1

Address: 2217Ch-2217Fh
Name: Watchdog Counter Threshold
ShortName: PR_CTR_THRSH_BCSUNIT

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Counter logic Threshold</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00150000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: U32</td>
</tr>
</tbody>
</table>

This field specifies the threshold that the hardware checks against for the value of the render clock counter before generating an interrupt. The counter in hardware generates an interrupt when the threshold is reached, rolls over and starts counting again. The interrupt generated is the "Media Hang Notify" interrupt since this watchdog timer is intended primarily to remedy VLD hangs on the main pipeline.
### Reported BitRateControl parameter Mask

**Register:** MMIO: 0/2/0  
**Project:** CHV, BSW  
**Source:** VideoCS  
**Default Value:** 0x00000000  
**Access:** RO  
**Size (in bits):** 32  
**Trusted Type:** 1  
**Address:** 12900h  
**Valid Projects:** CHV, BSW

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:6</td>
<td><strong>Reserved</strong></td>
<td>MBZ</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td><strong>Final Bitstream Buffer Overrun Mask</strong></td>
<td>U1</td>
<td>This is same bit reflected in MFX_VP8_ENC_CFG DW2 bit5. This denotes Final bitstream buffer overrun feature is enabled.</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td><strong>Intermediate Bitstream Buffer Overrun Mask</strong></td>
<td>U1</td>
<td>This is same bit reflected in MFX_VP8_ENC_CFG DW2 bit4. This denotes intermediate bitstream buffer overrun feature is enabled.</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td><strong>Intra MB Bit Count Conformance Mask</strong></td>
<td>U1</td>
<td>This is legacy support as AVC for Intra MB Bit Count conformance. This is same bit reflected in MFX_VP8_ENC_CFG DW2 bit3. This feature is not validated.</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td><strong>Inter MB Bit Count Conformance Mask</strong></td>
<td>U1</td>
<td>This is legacy support as AVC for Inter MB Bit Count conformance. This is same bit reflected in MFX_VP8_ENC_CFG DW2 bit2. This feature is not validated.</td>
</tr>
</tbody>
</table>
### MFX_VP8_CNTRL_MASK - Reported BitRateControl parameter Mask

<table>
<thead>
<tr>
<th></th>
<th>Frame Bit Rate Overflow Mask</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><strong>Format:</strong> U1</td>
</tr>
<tr>
<td></td>
<td>This is same bit reflected in MFX_VP8_ENC_CFG DW2 bit0. It denotes if Frame Bit Rate Overflow is enabled for Bit Rate Control</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Frame Bit Rate Underflow Mask</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td><strong>Format:</strong> U1</td>
</tr>
<tr>
<td></td>
<td>This is same bit reflected in MFX_VP8_ENC_CFG DW2 bit1. It denotes if Frame Bit Rate Underflow is enabled for Bit Rate Control</td>
</tr>
</tbody>
</table>
Reported BitRateControl parameter Status

MFX_VP8_CNTRL_STATUS - Reported BitRateControl parameter Status

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>VideoCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>RO</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Trusted Type:</td>
<td>1</td>
</tr>
<tr>
<td>Address:</td>
<td>12904h</td>
</tr>
<tr>
<td>Valid Projects:</td>
<td>CHV, BSW</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:8</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td><strong>Qindex ClampHigh Status</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: U1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This denotes if Qindex is clamped by Qindex ClampHigh value programmed in MFX_VP8_PIC_STATE.DW7.</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td><strong>Qindex ClampLow Status</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: U1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This denotes if Qindex is clamped by Qindex ClampLow value programmed in MFX_VP8_PIC_STATE.DW7.</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td><strong>Final Bitstream Buffer Overrun Status</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: U1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This denotes if Final bitstream buffer overrun.</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td><strong>Intermediate Bitstream Buffer Overrun Status</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: U1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This denotes if any of the Intermediate bitstream buffer overrun. (including FrameHeader, Partition1 to Partition8)</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td><strong>Intra MB Bit Count Conformance Status</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: U1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This is legacy support as AVC for Intra MB Bit Count conformance. It denotes if Intra MB Bit Count meets conformance size. This feature is not validated.</td>
</tr>
</tbody>
</table>
### MFX_VP8_CNTRL_STATUS - Reported BitRateControl parameter Status

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
<th>Format</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Inter MB Bit Count Conformance Status</td>
<td>U1</td>
<td>This is legacy support as AVC for Inter MB Bit Count conformance. It denotes if Inter MB Bit Count meets conformance size. This feature is not validated.</td>
</tr>
<tr>
<td>1</td>
<td>Frame Bit Rate Overflow Status</td>
<td>U1</td>
<td>It denotes if Frame Bit Rate Overflow in current frame</td>
</tr>
<tr>
<td>0</td>
<td>Frame Bit Rate Underflow Status</td>
<td>U1</td>
<td>It denotes if Frame Bit Rate Underflow in current frame</td>
</tr>
</tbody>
</table>
Reported Bitstream Output Bit Count for Syntax Elements Only Register

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>VideoCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>RO</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Trusted Type:</td>
<td>1</td>
</tr>
<tr>
<td>Address:</td>
<td>128A4h</td>
</tr>
<tr>
<td>Valid Projects:</td>
<td>CHV, BSW</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>MFC Bitstream Syntax Element Only Bit Count</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Total number of bits in the bitstream output due to syntax elements only. It includes the data bytes only. This count is updated for every time the internal bitstream counter is incremented and its reset at image start.</td>
</tr>
</tbody>
</table>
Reported Bitstream Output Byte Count per Frame Register

**MFC_BITSTREAM_BYTECOUNT_FRAME - Reported Bitstream Output Byte Count per Frame Register**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>VideoCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>RO</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Trusted Type:</td>
<td>1</td>
</tr>
<tr>
<td>Address:</td>
<td>128A0h</td>
</tr>
<tr>
<td>Valid Projects:</td>
<td>CHV, BSW</td>
</tr>
</tbody>
</table>

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

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

<table>
<thead>
<tr>
<th>MFC_AVC_CABAC_BIN_COUNT_FRAME - Reported Bitstream Output CABAC Bin Count Register</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: VideoCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Access: RO</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Trusted Type: 1</td>
</tr>
<tr>
<td>Address: 128A8h</td>
</tr>
<tr>
<td>Valid Projects: CHV, BSW</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>MFC AVC Cabac Bin Count</strong></td>
</tr>
</tbody>
</table>

Total number of BINs in the bitstream output per frame from the encoder. This count is updated for every time the bin counter is incremented and its reset at image start.
Reported Final Bitstream Byte Count

<table>
<thead>
<tr>
<th>MFX_VP8_FRM_BYTE_CNT - Reported Final Bitstream Byte Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: VideoCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Access: RO</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Trusted Type: 1</td>
</tr>
<tr>
<td>Address: 12908h</td>
</tr>
<tr>
<td>Valid Projects: CHV, BSW</td>
</tr>
<tr>
<td>This register stores the count of bytes of the bitstream output per frame</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>Final BitStream Byte Count</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: U32</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This register contains Final Bitstream byte count</td>
</tr>
</tbody>
</table>
## Reported Frame Zero Padding Byte Count

<table>
<thead>
<tr>
<th>MFX_VP8_FRM_ZERO_PAD - Reported Frame Zero Padding Byte Count</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>VideoCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>RO</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Trusted Type:</td>
<td>1</td>
</tr>
<tr>
<td>Address:</td>
<td>1290Ch</td>
</tr>
<tr>
<td>Valid Projects:</td>
<td>CHV, BSW</td>
</tr>
</tbody>
</table>

This register stores Frame Zero Padding Byte Count

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
<tr>
<td>15:0</td>
<td></td>
<td><strong>Frame Zero Padding Byte Count</strong></td>
</tr>
<tr>
<td></td>
<td>15:0</td>
<td>Format: U16</td>
</tr>
</tbody>
</table>

This register contains Frame Zero Padding byte count. This is legacy support. This feature is not validated.
# Reported Timestamp Count

## TIMESTAMP - Reported Timestamp Count

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>RenderCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000h, 00000000 CHV, BSW</td>
</tr>
<tr>
<td>Access:</td>
<td>RO. This register is not set by the context restore.</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>64</td>
</tr>
<tr>
<td>Address:</td>
<td>02358h</td>
</tr>
<tr>
<td>Address:</td>
<td>12358h-1235Fh</td>
</tr>
<tr>
<td>Name:</td>
<td>Reported Timestamp Count</td>
</tr>
<tr>
<td>ShortName:</td>
<td>TIMESTAMP_VCSUNIT0</td>
</tr>
<tr>
<td>Address:</td>
<td>1A358h-1A35Fh</td>
</tr>
<tr>
<td>Name:</td>
<td>Reported Timestamp Count</td>
</tr>
<tr>
<td>ShortName:</td>
<td>TIMESTAMP_VECSUNIT</td>
</tr>
<tr>
<td>Address:</td>
<td>1C358h-1C35Fh</td>
</tr>
<tr>
<td>Name:</td>
<td>Reported Timestamp Count</td>
</tr>
<tr>
<td>ShortName:</td>
<td>TIMESTAMP_VCSUNIT1</td>
</tr>
<tr>
<td>Address:</td>
<td>22358h-2235Fh</td>
</tr>
<tr>
<td>Name:</td>
<td>Reported Timestamp Count</td>
</tr>
<tr>
<td>ShortName:</td>
<td>TIMESTAMP_BCSUNIT</td>
</tr>
</tbody>
</table>

## Description

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

Note: On Core platforms, the TIMESTAMP register is initialized with the value of the PCU ART and hence tracks bits 38:3 of the 100 MHz ART fairly closely. However, due to variability in the actual time it takes to download the ART value to GT, the value of the TIMESTAMP register will be lower than the value of the PCU ART by an amount dependent on the relative IA/CLR/GT frequencies at the time the timestamp was downloaded to GT (expected to range between 100 and 600 ns). When comparing the value sampled from this register by GT HW to values read from the PCU timer by other system agents, timing differences between GT HW reading the TIMESTAMP register and the involved non-GT agent(s) reading the PCU ART must also be comprehended.
### TIMESTAMP - Reported Timestamp Count

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>63:36</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format:</strong> MBZ</td>
</tr>
<tr>
<td>35:0</td>
<td>Timestamp Value</td>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format:</strong> U36</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>This register toggles based on time stamp granularity (base unit) defined in the “Time Stamp Bases” subsection in Power Management chapter.</td>
</tr>
</tbody>
</table>
## Reset Flow Control Messages

### RSTFCTLMSG - Reset Flow Control Messages

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>08108h</td>
</tr>
</tbody>
</table>

#### Soft-Reset and FLR Flow Control Message Registers

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td><strong>Message Mask</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Message Mask</td>
</tr>
<tr>
<td></td>
<td></td>
<td>In order to write to bits 15:0, the corresponding message mask bits must be written. For example, for bit 14 to be set, bit 30 needs to be 1 : 40004000</td>
</tr>
<tr>
<td>15:12</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td><strong>MEDIA 1 Reset flow acknowledgement message</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PM Acknowledgement Messages for Media 1 reset:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1': PREP_RST_MEDIA1_ACK</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Acknowledgement that graphics media1 (or 2nd vbox) is prepared for reset assertion.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0': DONE_MEDIA1_RST_ACK</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Acknowledgement that graphics media1 (or 2nd vbox) reset is de-asserted</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td><strong>WIDI Reset flow acknowledgement message</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PM Acknowledgement Messages for WIDI reset:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1': PREP_RST_WIDI_ACK</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Acknowledgement that graphics widi is prepared for reset assertion.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0': DONE_WIDI_RST_ACK</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Acknowledgement that graphics widi reset is de-asserted</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td><strong>Vebox Reset flow Acknowledge Message</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PM Acknowledgement Messages for Vebox reset:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1': PREP_RST_VEBOX_ACK</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Acknowledgement that graphics VE is prepared for reset assertion.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0': DONE_VEBOX_RST_ACK</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Acknowledgement that graphics VE reset is de-asserted</td>
</tr>
</tbody>
</table>
### RSTFCTLMSG - Reset Flow Control Messages

<table>
<thead>
<tr>
<th>7</th>
<th>Blitter Reset Flow Acknowledgement Messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>PM Acknowledgement Messages for Blitter reset:</td>
<td></td>
</tr>
<tr>
<td>'1' : PREP_RST_BLIT_ACK</td>
<td></td>
</tr>
<tr>
<td>- Acknowledgement that graphics blitter is prepared for reset assertion.</td>
<td></td>
</tr>
<tr>
<td>'0' : DONE_BLIT_RST_ACK</td>
<td></td>
</tr>
<tr>
<td>- Acknowledgement that graphics blitter reset is de-asserted</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>6</th>
<th>Media Reset Flow Acknowledgement Messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>PM Acknowledgement Messages for Media reset:</td>
<td></td>
</tr>
<tr>
<td>'1' : PREP_RST_MEDIA_ACK</td>
<td></td>
</tr>
<tr>
<td>- Acknowledgement that graphics media block is prepared for reset assertion.</td>
<td></td>
</tr>
<tr>
<td>'0' : DONE_MEDIA_RST_ACK</td>
<td></td>
</tr>
<tr>
<td>- Acknowledgement that the graphics media reset is de-asserted</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>5</th>
<th>Render Reset Flow Acknowledgement Messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>PM Acknowledgement Messages for Render reset:</td>
<td></td>
</tr>
<tr>
<td>'1' : PREP_RST_RENDER_ACK</td>
<td></td>
</tr>
<tr>
<td>- Acknowledgement that the graphics render block is prepared for reset assertion.</td>
<td></td>
</tr>
<tr>
<td>'0' : DONE_RENDER_RST_ACK</td>
<td></td>
</tr>
<tr>
<td>- Acknowledgement that the graphics render reset is de-asserted</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>4</th>
<th>GTI-Device Reset Flow Acknowledgement Messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>PM Acknowledgement Messages for GTI-Device reset:</td>
<td></td>
</tr>
<tr>
<td>'1' : PREP_RST_GTIDEV_ACK</td>
<td></td>
</tr>
<tr>
<td>- Acknowledgement that the GTI device is prepared for reset assertion.</td>
<td></td>
</tr>
<tr>
<td>'0' : DONE_GTIDEV_RST_ACK</td>
<td></td>
</tr>
<tr>
<td>- Acknowledgement that the GTI device reset is de-asserted</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>3</th>
<th>Reserved</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access:</td>
<td>RO</td>
</tr>
<tr>
<td>Reserved</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>2</th>
<th>FLR Done ack from Pmunit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access:</td>
<td>R/W Set</td>
</tr>
<tr>
<td>FLR Done ack from Pmunit:</td>
<td></td>
</tr>
<tr>
<td>1: PM unit sets this bit to acknowledge the FLR done message has been forwarded to SA through GAM interface.</td>
<td></td>
</tr>
<tr>
<td>0: Default Value. If the bit was set by PM then Cpunit hardware clears it once FLR is completed.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>RSTFCTLMSG - Reset Flow Control Messages</strong></td>
</tr>
<tr>
<td>---</td>
<td>-------------------------------------</td>
</tr>
<tr>
<td>1</td>
<td><strong>Global Resource Arbitration Acknowledgement Messages</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td>Global Resource Arbitration Acknowledgement Message from PM:</td>
</tr>
<tr>
<td></td>
<td>'1' : CP_ARB_REQ_ACK - Acknowledgement for CPunit's global resource arbitration request</td>
</tr>
<tr>
<td></td>
<td>'0' : CP_ARB_RELEASE_ACK - Acknowledgement to CPunit's release of global resources</td>
</tr>
<tr>
<td>0</td>
<td><strong>CP Busy / Idle Status Acknowledgement Messages</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td>CP Busy / Idle Status Acknowledgement Message from PM:</td>
</tr>
<tr>
<td></td>
<td>'0' : CP_NOT_BUSY_ACK - Acknowledgement that the CPunit is idle.</td>
</tr>
<tr>
<td></td>
<td>'1' : CP_BUSY_ACK - Acknowledgement that the CPunit is busy.</td>
</tr>
</tbody>
</table>
## RESET Messaging Register for Clocking Unit

<table>
<thead>
<tr>
<th>MSG_RESET_GCP - RESET Messaging Register for Clocking Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Address:</strong> 08030h</td>
</tr>
</tbody>
</table>

Message registers have bit-wise masking applied for writes. The register consists of 16 bits of data in [15:0], and 16 bits of corresponding masks in [31:16]. In order to set bit0, for example, the data would be 0x0001_0001. In order to clear bit0, for example, the data would be 0x0001_0000. Note that mask bit is the data bit offset + 16. Message registers are protected from non-GT writes via the Message Channel.

**Request to Prepare for Reset**
1'b0 : Reset complete <default>  
1'b1 : Prepare for reset

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>15:8</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>Request to Prepare for FLR</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>[7] Prepare for devrst_b Domain Reset (FLR)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note: All resets except busrst_b will be asserted for an FLR</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>Request to Prepare for Media1 Reset</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>[6] Prepare for cmrst_b Domain Reset (vcs1unit)</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>Request to Prepare for Wi-Di Reset</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>[5] Prepare for cwrst_b Domain Reset (winunit)</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>Request to Prepare for Blitter Reset</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>[3] Prepare for crblitrst_b Domain Reset (bcsunit)</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>Request to Prepare for VEBox Reset</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>[2] Prepare for cvrst_b Domain Reset (vecsunit)</td>
</tr>
</tbody>
</table>
### MSG_RESET_GCP - RESET Messaging Register for Clocking Unit

<table>
<thead>
<tr>
<th></th>
<th>Request to Prepare for Media0 Reset</th>
<th>Request to Prepare for Render Reset</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><strong>Access:</strong> R/W</td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td>[1] Prepare for cmrst_b Domain Reset (vcs0unit)</td>
<td>[0] Prepare for crrst_b Domain Reset (csunit)</td>
</tr>
</tbody>
</table>
# Resource Streamer Context Offset

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>RenderCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00003B00</td>
</tr>
<tr>
<td>Access:</td>
<td>Read/32 bit Write Only</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>021B4h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:6</td>
<td><strong>RS Offset</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>U26</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This field indicates the offset (64 bytes granular) in to the logical rendering context to which Resource Streamer context is save/restored when enabled. This field register must not be written directly (via MMIO) unless the Command Streamer is completely idle (i.e., the Ring Buffer is empty and the pipeline is idle) and RC6 is disabled. On way to program this register is via Load Register Immediate command in the ring buffer as part of initialization sequence.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECh</td>
<td>[Default]</td>
<td>DefaultValueDesc</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>5:0</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MBZ</td>
</tr>
</tbody>
</table>
Resource Streamer Preemption Status

**RS_PREEMPT_STATUS - Resource Streamer Preemption Status**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>RenderCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0215Ch</td>
</tr>
</tbody>
</table>

**Preemption from First Level Batch Buffer:** This register contains the offset in to the Batch Buffer on which Resource streamer got preempted. Note that it is offset from the Batch Start Address and not the graphics address corresponding to the preempted instruction on Batch Buffer. This register's contents will be looked at by Command Streamer on the next RS Start Trigger to provide the appropriate batch start address. Following preemption, context save happens on which this register is context saved, after context save the register gets auto reset by Command Streamer. This register gets context save/restored by Render Command Streamer as part of its render context. **Preemption from Second Level Batch Buffer:** This register contains the graphics address of the instruction in Second Level Batch Buffer on which Resource streamer got preempted. This register's contents will be looked at by Command Streamer on the next RS Start Trigger to provide the appropriate batch start address. Following preemption, context save happens on which this register is context saved, after context save the register gets auto reset by Command Streamer. This register gets context save/restored by Render Command Streamer as part of its render context.

**Programming Notes**

- This register is accessed by Render Command Streamer as part of render context save/restore; this register should be exercised by S/W only for resetting the register contents if required.
- Following preemption if there is no context save, SW should program this register with 0x0 so that it does not interfere with proceeding workloads.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:2</td>
<td><strong>Batch Buffer Offset</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Offset[31:2]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This field specifies the DWord-aligned offset from the batch start address on which Resource Streamer got preempted.</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td><strong>RS_PREEMPT_STATUS</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MBZ</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This field when not set indicates RS got preempted on a natural sync point else it got preempted on a draw call.</td>
</tr>
<tr>
<td>RS_PREEMPT_STATUS - Resource Streamer Preemption Status</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>RS_PREEMPTED</td>
<td></td>
</tr>
<tr>
<td>Default Value:</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Format:</td>
<td>Enable</td>
<td></td>
</tr>
</tbody>
</table>

If this bit is set indicates Resource Streamer got preempted. Other fields of this register are valid only when this bit is set.
## Restored Timestamp LSDW

### RTSLSDW - Restored Timestamp LSDW

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0A530h-0A533h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Restored Time Stamp Storage</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Restored Time Stamp Storage</td>
</tr>
</tbody>
</table>


### RTSMSDW - Restored timestamp MSDW

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0A534h-0A537h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Restored Time Stamp Storage</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Restored Time Stamp Storage</td>
</tr>
</tbody>
</table>
RID_CC

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>BASE_CLASS_CODE</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>23:16</td>
<td><strong>SUB_CLASS_CODE</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>15:8</td>
<td><strong>PROGRAMMING_INTERFACE</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default Value: 00h</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access: RO</td>
<td></td>
</tr>
<tr>
<td>7:0</td>
<td><strong>REVISION_ID</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default Value: 00000000b</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
<td></td>
</tr>
</tbody>
</table>

RID_CC - RID_CC

Register Space: PCI: 0/2/0
Project: CHV, BSW
Source: PRM
Default Value: 0x00000000
Size (in bits): 32

Address: 00008h

Revision Identification and Class code register. SOXi Context Save/Restore: Yes.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>BASE_CLASS_CODE</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>23:16</td>
<td><strong>SUB_CLASS_CODE</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>15:8</td>
<td><strong>PROGRAMMING_INTERFACE</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default Value: 00h</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access: RO</td>
<td></td>
</tr>
<tr>
<td>7:0</td>
<td><strong>REVISION_ID</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default Value: 00000000b</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
<td></td>
</tr>
</tbody>
</table>

RID: The value in this field reflects the value of strapRID[7:0] (which is an input pin of GVD).
For VV: The reset value will be the same as MID bits 23:16 (reference the MID register for MID register details).
This register is read-able by any agent.
Under SAI protection, the PMC and Punit (ie. ‘Trusted_FW’) can write this register and change the default value. No other agent has the ability to update this register. The PMC implementation will consider SRID and CRID before updating this register. Any PMC updates will be prior to Device 2 configuration.
Note: The MID register will always reflect the stepping information. Even if PMC updates this register, the MID is available as a SRID reference.
## RING_BUFFER_HEAD_PREEMPT_REG

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

### Preemptable Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>MI_ARB_CHECK</td>
<td>RenderCS</td>
</tr>
<tr>
<td>3D_PRIMITIVE</td>
<td>RenderCS</td>
</tr>
<tr>
<td>GPGPU_WALKER</td>
<td>RenderCS</td>
</tr>
<tr>
<td>MEDIA_STATE_FLUSH</td>
<td>RenderCS</td>
</tr>
<tr>
<td>Pipe_CONTROL (Only in GPGPU mode of pipeline selection)</td>
<td>RenderCS</td>
</tr>
</tbody>
</table>

### Register Details

- **Register Space**: MMIO: 0/2/0
- **Project**: CHV, BSW
- **Source**: PRM
- **Default Value**: 0x00000000
- **Access**: R/W
- **Size (in bits)**: 32

#### Address: 0214Ch-0214Fh
- **Name**: RING_BUFFER_HEAD_PREEMPT_REG
- **ShortName**: RING_BUFFER_HEAD_PREEMPT_REG_RCSUNIT

#### Address: 1214Ch-1214Fh
- **Name**: RING_BUFFER_HEAD_PREEMPT_REG
- **ShortName**: RING_BUFFER_HEAD_PREEMPT_REG_VCSUNIT0

#### Address: 1A14Ch-1A14Fh
- **Name**: RING_BUFFER_HEAD_PREEMPT_REG
- **ShortName**: RING_BUFFER_HEAD_PREEMPT_REG_VCSUNIT1

#### Address: 2214Ch-2214Fh
- **Name**: RING_BUFFER_HEAD_PREEMPT_REG
- **ShortName**: RING_BUFFER_HEAD_PREEMPT_REG_BCSUNIT
### RING_BUFFER_HEAD_PREEMPT_REG - RING_BUFFER_HEAD_PREEMPT_REG

- MI_ATOMIC (Post Sync Operation set in GPGPU mode of pipeline selection)
- MI_SEMAPHORE_SIGNAL (Post Sync Operation set in GPGPU mode of pipeline selection)

### Programming Notes

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:21</td>
<td>Last Wrap Count</td>
</tr>
<tr>
<td></td>
<td>20:2</td>
<td>Preempted Head Offset</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: U19</td>
</tr>
</tbody>
</table>

This field contains the Head pointer offset in the ring when the last MI_ARB_CHECK command was executed and caused the head pointer to move due to the UHPTR register being valid.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Ring/Batch Indicator</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1:0</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: Enabled</td>
</tr>
</tbody>
</table>

### Value | Name          | Description                                                                 |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>Ring</td>
<td>Preemptable command was executed in ring and caused head pointer to be updated.</td>
</tr>
<tr>
<td>1h</td>
<td>Batch</td>
<td>Preemptable command was executed in batch and caused head pointer to be updated.</td>
</tr>
<tr>
<td>2h</td>
<td>2nd level batch</td>
<td>Preemptable command was executed in second level batch and caused head pointer to be updated.</td>
</tr>
</tbody>
</table>
Ring Buffer Control

RING_BUFFER_CTL - Ring Buffer Control

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
</tbody>
</table>

Address: 0203Ch-0203Fh
Name: Ring Buffer Control
ShortName: RING_BUFFER_CTL_RCSUNIT

Address: 1203Ch-1203Fh
Name: Ring Buffer Control
ShortName: RING_BUFFER_CTL_VCSUNIT0

Address: 1A03Ch-1A03Fh
Name: Ring Buffer Control
ShortName: RING_BUFFER_CTL_VSCUNIT

Address: 1C03Ch-1C03Fh
Name: Ring Buffer Control
ShortName: RING_BUFFER_CTL_VCSUNIT1

Address: 2203Ch-2203Fh
Name: Ring Buffer Control
ShortName: RING_BUFFER_CTL_BCSUNIT

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

**Ring Buffer Head and Tail Offsets must be properly programmed before it is enabled. A Ring Buffer can be enabled when empty.**

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:21</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
<tr>
<td>20:12</td>
<td></td>
<td>Buffer Length</td>
</tr>
</tbody>
</table>

Source: RenderCS, BlitterCS, VideoCS, VideoEnhancementCS
## RING_BUFFER_CTL - Ring Buffer Control

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1 page = 4 KB</td>
<td></td>
</tr>
<tr>
<td>1FFh</td>
<td>512 pages = 2 MB</td>
<td></td>
</tr>
</tbody>
</table>

| 11    | RBWait                | Description: Indicates that this ring has executed a WAIT_FOR_EVENT instruction and is currently waiting. Software can write a "1" to clear this bit, write of "0" has no effect. When the RB is waiting for an event and this bit is cleared, the wait will be terminated and the RB will be returned to arbitration. RenderCS: RBWait is not set on executing WAIT_FOR_EVENT instruction waiting on Async Flip Pending. |

| 10    | Semaphore Wait        | Description: Indicates that this ring has executed a MI_SEMAPHORE_WAIT instruction and is currently waiting for wait condition to satisfy. |

| 9     | Reserved              | Format: MBZ                                                                                           |

| 8     | Reserved              | Project: CHV, BSW                                                                                     |

| 7:3   | Reserved              | Format: MBZ                                                                                           |

| 2:1   | Automatic Report Head Pointer | Project: CHV, BSW                                                                                     |

| Source | PRM                         | Description: This field is written by software to control the automatic reporting (write) of this ring buffer's Head Pointer register (register DWord 1) to the corresponding location within the Hardware Status Page. Automatic reporting can either be disabled or enabled at 4KB, 64KB or 128KB boundaries within the ring buffer. When Execlist Enable bit is set the head pointer will be reported to the head pointer location in the Per-Process Hardware Status Page. MI_AUTOREPORT_4KB option is not supported on A stepping. |
## RING BUFFER_CTL - Ring Buffer Control

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

### Ring Buffer Enable

Format: Enable

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

### Programming Notes

**Source:** RenderCS

Ring Buffer Mode of Scheduling: SW must follow the below programming notes during SW initialization phase or while enabling render engine’s ring buffer for the first time, this would be coming out of boot, standby, hibernate or reset. SW must set the Force Wakeup bit to prevent GT from entering C6.

- SW must dispatch workload (dummy context) to initialize render engine with default state such that any context switches that occur subsequently (Power Save) will save and restore coherent device state. Indirect pointers used in 3D states must point to valid graphics surface existing in memory. PP_DCLV followed by PP_DIR_BASE register should be programmed as part of initialization workload if PPGTT is enabled in GFX_MODE register.
- SW must ensure all the register (MMIO) initialization/programming through CPU happens in this block or latter, this ensures the MMIO state is save/restored on subsequent context switches (Power Sequences).
- Once the render engine is programmed with valid state and the configuration, Force Wakeup bit should be reset to enable C6 entry.

**Source:** RenderCS

Render CS Only: Ring Buffer Mode of Scheduling: SW must follow the below programming notes before disabling ring buffer to ensure HW is not in middle of the IDLE flows.

- SW must set the Force Wakeup bit to prevent GT from entering C6.
- Disable IDLE messaging in CS (Write 0x2050[31:0] = 0x00010000)
- Poll/Wait for register bits of 0x22AC[6:0] turn to 0x30 value.
- Disable Ring Buffer
- Enable IDLE messaging in CS (Write 0x2050[31:0] = 0x00010000)
- Force Wakeup bit should be reset to enable C6 entry.
Ring Buffer Current Context ID Register

<table>
<thead>
<tr>
<th>BCS_RCCID - Ring Buffer Current Context ID Register</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: BlitterCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000, 0x00000000</td>
</tr>
<tr>
<td>Access: R/W</td>
</tr>
<tr>
<td>Size (in bits): 64</td>
</tr>
<tr>
<td>Address: 22190h-22197h</td>
</tr>
</tbody>
</table>

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

**Programming Notes**

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>63:0</td>
<td><strong>Unnamed</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>See Context Descriptor for BCS.</td>
</tr>
</tbody>
</table>
## Ring Buffer Head

<table>
<thead>
<tr>
<th><strong>RING_BUFFER_HEAD - Ring Buffer Head</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 02034h-02037h</td>
</tr>
<tr>
<td><strong>Name:</strong> Ring Buffer Head</td>
</tr>
<tr>
<td><strong>ShortName:</strong> RING_BUFFER_HEAD_RCSUNIT</td>
</tr>
<tr>
<td><strong>Address:</strong> 12034h-12037h</td>
</tr>
<tr>
<td><strong>Name:</strong> Ring Buffer Head</td>
</tr>
<tr>
<td><strong>ShortName:</strong> RING_BUFFER_HEAD_VCSUNIT0</td>
</tr>
<tr>
<td><strong>Address:</strong> 1A034h-1A037h</td>
</tr>
<tr>
<td><strong>Name:</strong> Ring Buffer Head</td>
</tr>
<tr>
<td><strong>ShortName:</strong> RING_BUFFER_HEAD_VECSUNIT</td>
</tr>
<tr>
<td><strong>Address:</strong> 1C034h-1C037h</td>
</tr>
<tr>
<td><strong>Name:</strong> Ring Buffer Head</td>
</tr>
<tr>
<td><strong>ShortName:</strong> RING_BUFFER_HEAD_VCSUNIT1</td>
</tr>
<tr>
<td><strong>Address:</strong> 22034h-22037h</td>
</tr>
<tr>
<td><strong>Name:</strong> Ring Buffer Head</td>
</tr>
<tr>
<td><strong>ShortName:</strong> RING_BUFFER_HEAD_BCSUNIT</td>
</tr>
</tbody>
</table>

### Description

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

<table>
<thead>
<tr>
<th><strong>DWord</strong></th>
<th><strong>Bit</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:21</td>
<td><strong>Wrap Count</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format:</strong> U11 count of ring buffer wraps</td>
</tr>
</tbody>
</table>

This field is incremented by 1 whenever the **Head Offset** wraps from the end of the buffer back to the start (i.e., whenever it wraps back to 0). Appending this field to the **Head Offset** field effectively creates a virtual 4GB Head "Pointer" which can be used as a tag associated with instructions placed in a ring buffer. The Wrap Count itself will wrap to 0 upon overflow.
## RING_BUFFER_HEAD - Ring Buffer Head

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Head Offset</td>
<td>GraphicsAddress[20:2] DWord Offset</td>
</tr>
</tbody>
</table>
| | This field indicates the offset of the next instruction DWord to be parsed. Software will initialize this field to select the first DWord to be parsed once the RB is enabled. (Writing the Head Offset while the RB is enabled is UNDEFINED). Subsequently, the device will increment this offset as it executes instructions - until it reaches the QWord specified by the Tail Offset. At this point the ring buffer is considered “empty”.

| Programming Notes | |
| A RB can be enabled empty or containing some number of valid instructions. |

| 1 | Reserved |
| Format: | MBZ |

| 0 | Reserved |
| Project: | CHV, BSW |
| Format: | MBZ |
Ring Buffer Start

<table>
<thead>
<tr>
<th>RING_BUFFER_START - Ring Buffer Start</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 02038h-0203Bh</td>
</tr>
<tr>
<td><strong>Name:</strong> Ring Buffer Start</td>
</tr>
<tr>
<td><strong>ShortName:</strong> RING_BUFFER_START_RCSUNIT</td>
</tr>
<tr>
<td><strong>Address:</strong> 12038h-1203Bh</td>
</tr>
<tr>
<td><strong>Name:</strong> Ring Buffer Start</td>
</tr>
<tr>
<td><strong>ShortName:</strong> RING_BUFFER_START_VCSUNIT0</td>
</tr>
<tr>
<td><strong>Address:</strong> 1A038h-1A03Bh</td>
</tr>
<tr>
<td><strong>Name:</strong> Ring Buffer Start</td>
</tr>
<tr>
<td><strong>ShortName:</strong> RING_BUFFER_START_VECSUNIT</td>
</tr>
<tr>
<td><strong>Address:</strong> 1C038h-1C03Bh</td>
</tr>
<tr>
<td><strong>Name:</strong> Ring Buffer Start</td>
</tr>
<tr>
<td><strong>ShortName:</strong> RING_BUFFER_START_VCSUNIT1</td>
</tr>
<tr>
<td><strong>Address:</strong> 22038h-2203Bh</td>
</tr>
<tr>
<td><strong>Name:</strong> Ring Buffer Start</td>
</tr>
<tr>
<td><strong>ShortName:</strong> RING_BUFFER_START_BCSUNIT</td>
</tr>
</tbody>
</table>

**Description**

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

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

Doc Ref # IHD-OS-CHV-BSW-Vol 2c- 10.15 1131
# Ring Buffer Tail

<table>
<thead>
<tr>
<th>RING_BUFFER_TAIL - Ring Buffer Tail</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Access: R/W</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 02030h-02033h</td>
</tr>
<tr>
<td>Name: Ring Buffer Tail</td>
</tr>
<tr>
<td>ShortName: RING_BUFFER_TAIL_RCSUNIT</td>
</tr>
<tr>
<td>Address: 12030h-12033h</td>
</tr>
<tr>
<td>Name: Ring Buffer Tail</td>
</tr>
<tr>
<td>ShortName: RING_BUFFER_TAIL_VCSUNIT0</td>
</tr>
<tr>
<td>Address: 1A030h-1A033h</td>
</tr>
<tr>
<td>Name: Ring Buffer Tail</td>
</tr>
<tr>
<td>ShortName: RING_BUFFER_TAIL_VECSUNIT</td>
</tr>
<tr>
<td>Address: 1C030h-1C033h</td>
</tr>
<tr>
<td>Name: Ring Buffer Tail</td>
</tr>
<tr>
<td>ShortName: RING_BUFFER_TAIL_VCSUNIT1</td>
</tr>
<tr>
<td>Address: 22030h-22033h</td>
</tr>
<tr>
<td>Name: Ring Buffer Tail</td>
</tr>
<tr>
<td>ShortName: RING_BUFFER_TAIL_BCSUNIT</td>
</tr>
</tbody>
</table>

## Description

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:21</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
</tbody>
</table>

---

```
Doc Ref # IHD-OS-CHV-BSW-Vol 2c-10.15
```
## RING_BUFFER_TAIL - Ring Buffer Tail

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

### Programming Notes

- **[Ring Buffer Mode Of scheduling only][Video CS, Video Enhancement CS, Blitter CS]:** HW loses Page Directory (PPGTT) information on becoming IDLE. SW must always program the PD information through MI_LOAD_REGISTER_IMM command in the ring buffer prior to programming workload begins on every ring dispatch. This will ensure Page Directory information is not lost due to IDLE flows.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2:0</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td>Format:</td>
<td>MBZ</td>
</tr>
</tbody>
</table>
## Root Table Address Pointer Value First 31_0

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>First Address 31 to 0</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bits 31:11 = Root Table Address Pointer Value 31:11.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bits 10:1 = Reserved.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit 0 = Enabled for Root Table Address Pointer Value 31:11.</td>
</tr>
</tbody>
</table>
### Root Table Address Pointer Value Second 31_0

<table>
<thead>
<tr>
<th><strong>RTAPV_2_310 - Root Table Address Pointer Value Second 31_0</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 0F504h</td>
</tr>
</tbody>
</table>

This register is used to store local copy of the Root Table address pointer value.

<table>
<thead>
<tr>
<th><strong>DWord</strong></th>
<th><strong>Bit</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Second Address 31 to 0</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

Bits 31:8 = Reserved.

Bits 7:1 = Root Table Address Pointer Value 38:32.

Bit 0 = Enabled for Root Table Address Pointer Value 38:32.
# RP Decrease Limit

<table>
<thead>
<tr>
<th><strong>RPDECLIMIT - RP Decrease Limit</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 0A030h-0A033h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>23:0</td>
<td></td>
<td><strong>Decrease Threshold</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

**Decrease Threshold (DECLIMIT):**

This register contains the threshold used to determine whether a switch to a higher frequency is desirable. FRQDUM determines the meaning of this register:

- Busy Min Continuous Limit
- Busy Min Average Limit
- Idle Max Continuous Limit
- The values are:
  - 0 = 0 usec
  - 1 = 1.28 usec
  - 2 = 2.56 usec
  - 3 = 3.84 usec
  - FF FFFF = 21.474 sec
RP Down Timeout

<table>
<thead>
<tr>
<th>RPDNTIMOUT - RP Down Timeout</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x0000000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 0A010h-0A013h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: <strong>RO</strong></td>
</tr>
<tr>
<td>23:0</td>
<td></td>
<td><strong>Down Timeout</strong></td>
</tr>
<tr>
<td></td>
<td>Access: <strong>R/W</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>The value in this register contains the timeout value for gfx idleness due to stopped graphics clocks (CPD) necessary to trigger the Render Geyserville Downward Timeout interrupt.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 = 0 usec</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 = 1.28 usec</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2 = 2.56 usec</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3 = 3.84 usec</td>
<td></td>
</tr>
<tr>
<td></td>
<td>FF FFFF = 21.474 sec</td>
<td></td>
</tr>
</tbody>
</table>
# RP Downwards Evaluation Interval

## RPDNEI - RP Downwards Evaluation Interval

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0A06Ch-0A06Fh</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td>23:0</td>
<td></td>
<td><strong>Evaluation Interval Period for Downwards Freq Direction</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
</tbody>
</table>

The geyserville performance status will be averaged over this interval, and the results will be used to possibly recommend a switch to the slower render clock frequency (either directly by hardware or via an interrupt activating a sw decision).

- 0 = 0 usec
- 1 = 1.28 usec
- 2 = 2.56 usec
- 3 = 3.84 usec
- FF FFFF = 21.474 sec

pmcr_ei_down[23:0]
RP Increase Limit

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0A02Ch-0A02Fh</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Register: RPINCLIMIT - RP Increase Limit</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
</tr>
<tr>
<td><strong>Bit</strong></td>
</tr>
<tr>
<td>--------</td>
</tr>
<tr>
<td>31:24</td>
</tr>
<tr>
<td>23:0</td>
</tr>
</tbody>
</table>

Increase Threshold (INCLIMIT):
This register contains the threshold used to determine whether a switch to a higher frequency is desirable. FRQIUM determines the meaning of this register:

Busy Max Continuous Limit
Busy Max Average Limit
Idle Min Continuous Limit

The values are:
0 = 0 usec
1 = 1.28 usec
2 = 2.56 usec
3 = 3.84 usec
FF FFFF = 21.474 sec
## RP_NORMAL - RP Normal Software Frequency Request

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31</td>
<td><strong>Turbo Disable</strong>&lt;br&gt;Access: R/W&lt;br&gt;The Turbo Disable bit is determined by SW. NOTE: If Turbo is disable for ANY thread, it will prevent turbo for ALL threads.</td>
</tr>
<tr>
<td></td>
<td>30:24</td>
<td><strong>P State Request</strong>&lt;br&gt;Access: R/W&lt;br&gt;This field indicates the maximum P-State request in units of 100MHz.</td>
</tr>
<tr>
<td></td>
<td>23:18</td>
<td><strong>P State Offset</strong>&lt;br&gt;Access: R/W&lt;br&gt;This field defined the number of steps that Energy Efficient P-State is allowed to fall in units of 100 MHz.</td>
</tr>
<tr>
<td></td>
<td>17:14</td>
<td><strong>Energy Efficient policy</strong>&lt;br&gt;Access: R/W&lt;br&gt;The energy efficiency policy is determined by SW.</td>
</tr>
<tr>
<td></td>
<td>13:0</td>
<td><strong>Reserved</strong>&lt;br&gt;Access: RO</td>
</tr>
</tbody>
</table>
### RP Software Frequency Request Hysteresis

<table>
<thead>
<tr>
<th><strong>RPSWFREQHYST - RP Software Frequency Request Hysteresis</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 0A004h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
<th>Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:6</td>
<td><strong>Reserved</strong></td>
<td>RO</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>5:0</td>
<td></td>
<td><strong>RP SW Freq Request Hysteresis</strong></td>
<td>R/W</td>
<td></td>
</tr>
</tbody>
</table>

**Programming Notes**

Not supported. It must be 0 at all times.
### RP Upwards Evaluation Interval

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>23:0</td>
<td></td>
<td>Evaluation Interval Period for Upwards Freq Direction</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The geyserville performance status will be averaged over this interval, and the results will be used to possibly recommend a switch to the faster render clock frequency (either directly by hardware or via an interrupt activating a sw decision).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 = 0 usec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 = 1.28 usec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 = 2.56 usec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 = 3.84 usec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FFFFF = 21.474 sec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>pmcr_ei_up[23:0]</td>
</tr>
</tbody>
</table>
##RP Video Turbo Software Frequency Request

| **RP_FREQ_VIDEOTURBO** - RP Video Turbo Software Frequency Request |
|---|---|
| **Register Space:** | MMIO: 0/2/0 |
| **Project:** | CHV, BSW |
| **Source:** | PRM |
| **Default Value:** | 0x00000000 |
| **Size (in bits):** | 32 |
| **Address:** | 0A00Ch |

This 32 bit value is written to the PCU IO_THREAD_P_REQ register when in Video Turbo mode.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31</td>
<td><strong>Turbo Disable</strong>&lt;br&gt;<strong>Access:</strong> R/W&lt;br&gt;The Turbo Disable bit is determined by SW. NOTE: If Turbo is disable for ANY thread, it will prevent turbo for ALL threads.</td>
</tr>
<tr>
<td></td>
<td>30:24</td>
<td><strong>P State Request</strong>&lt;br&gt;<strong>Access:</strong> R/W&lt;br&gt;This field indicates the maximum P-State request in units of 100MHz.</td>
</tr>
<tr>
<td></td>
<td>23:18</td>
<td><strong>P State Offset</strong>&lt;br&gt;<strong>Access:</strong> R/W&lt;br&gt;This field defined the number of steps that Energy Efficient P-State is allowed to fall in units of 100 MHz.</td>
</tr>
<tr>
<td></td>
<td>17:14</td>
<td><strong>Energy Efficient policy</strong>&lt;br&gt;<strong>Access:</strong> R/W&lt;br&gt;The energy efficiency policy is determined by SW.</td>
</tr>
<tr>
<td></td>
<td>13:0</td>
<td><strong>Reserved</strong>&lt;br&gt;<strong>Access:</strong> RO</td>
</tr>
</tbody>
</table>
RS_PREEMPT_STATUS_UDW

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>RenderCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>02174h</td>
</tr>
</tbody>
</table>

**Preemption from First Level Batch Buffer:** This register contains the offset in to the Batch Buffer on which Resource streamer got preempted. Note that it is offset from the Batch Start Address and not the graphics address corresponding to the preempted instruction on Batch Buffer. This register’s contents will be looked at by Command Streamer on the next RS Start Trigger to provide the appropriate batch start address. Following preemption, context save happens on which this register is context saved, after context save the register gets auto reset by Command Streamer. This register gets context save/restored by Render Command Streamer as part of its render context.

**Preemption from Second Level Batch Buffer:** This register contains the graphics address of the instruction in Second Level Batch Buffer on which Resource streamer got preempted. This register’s contents will be looked at by Command Streamer on the next RS Start Trigger to provide the appropriate batch start address. Following preemption, context save happens on which this register is context saved, after context save the register gets auto reset by Command Streamer. This register gets context save/restored by Render Command Streamer as part of its render context.

### Programming Notes
- This register is accessed by Render Command Streamer as part of render context save/restore; this register should be exercised by S/W only for resetting the register contents if required.
- Following preemption if there is no context save, SW should program this register with 0x0 so that it does not interfere with proceeding workloads.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format:</strong> MBZ</td>
</tr>
<tr>
<td>15:0</td>
<td></td>
<td><strong>Batch Buffer Offset Upper DWORD</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format:</strong> GraphicsAddress[47:32]</td>
</tr>
</tbody>
</table>

This field specifies the 4GB aligned base address of gfx 4GB virtual address space within the host's 64-bit virtual address space of the last preempted second level batch buffer in resource streamer.
RS Preemption Hint

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>RenderCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>024C0h</td>
</tr>
</tbody>
</table>

This register contains the Dword aligned Graphics address in to the Batch Buffer corresponding to either 3D_PRIMITIVE or RCS-RS sync command called Preemption Hint Address. When Preemption Hint Address is enabled, RS will honor preemption request from RCS only on parsing 3D_PRIMITIVE/RCS-RS sync command at Preemption Hint Address.

### Programming Notes

**Programming Restriction:**
This register should NEVER be programmed in functional mode, this should be used only in validation mode to achieve deterministic behavior of preempting Resource Streamer in command stream. Programmer has to ensure that RS Preemption Hint register gets programmed well before RS gets preempted by RCS. Note that this register should be programmed with caution as it can lead to indefinite stalls in RS.

- a. RS will preempt on receiving preemption request from RCS only on reaching the instruction in the batch buffer corresponding to the address mentioned in RS_PREEMPT_HINT. RS could hit an RCS-RS sync command before reaching the address mentioned in the RS_PREEMPT_HINT, in this case RS should preempt on the sync command.
- b. RS could hit the address mentioned in 3D_PREEMPT_HINT before receiving preempt request from RCS. In this case RS will stall at this command until it receives preemption request from RCS and then preempts.

### DWord 0

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>31:2</td>
<td>Preemption Hint Address</td>
</tr>
<tr>
<td></td>
<td>Format: U30</td>
</tr>
<tr>
<td></td>
<td>This field contains the Dword aligned Graphics Address in to the batch buffer as Preemption Hint.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Reserved</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Preemption Hint</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: Enabled</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>Disabled</td>
<td>Preemption hint is disabled for Resource Streamer.</td>
</tr>
<tr>
<td>1h</td>
<td>Enabled</td>
<td>Preemption hint is enabled for Resource streamer.</td>
</tr>
</tbody>
</table>
RS Preemption Hint UDW

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>RenderCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>024C4h</td>
</tr>
</tbody>
</table>

This register contains the 4GB aligned base address of gfx 4GB virtual address space within the host's 64-bit virtual address space of the Batch Buffer corresponding to either 3D_PRIMITIVE or RCS-RS sync command called Preemption Hint Address. When Preemption Hint Address is enabled, RS will honor preemption request from RCS only on parsing 3D_PRIMITIVE/RCS-RS sync command at Preemption Hint Address.

**Restriction**

This register should NEVER be programmed in functional mode, this should be used only in validation mode to achieve deterministic behavior of preempting Resource Streamer in command stream. See RS_PRE_HINT definition for further restrictions.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
<tr>
<td>15:0</td>
<td></td>
<td><strong>Preemption Hint Address Upper DWORD</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: GraphicsAddress[47:32]</td>
</tr>
</tbody>
</table>

This field contains the 4GB aligned base address of gfx 4GB virtual address space within the host's 64-bit virtual address space of the batch buffer as Preemption Hint.
## Sampler control register

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td>ECO Reserved 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved: MBZ</td>
</tr>
<tr>
<td></td>
<td>15:8</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td>7:3</td>
<td>Sampler unit select</td>
</tr>
<tr>
<td>2</td>
<td>ECO Reserved 2</td>
<td></td>
</tr>
<tr>
<td>1:0</td>
<td>ECO Reserved 3</td>
<td></td>
</tr>
</tbody>
</table>
SAMPLER Mode Register

**SAMPLER_MODE - SAMPLER Mode Register**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>RenderCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Trusted Type:</td>
<td>1</td>
</tr>
</tbody>
</table>

Address: 07028h
Valid Projects: CHV, BSW

This register has bit-wise masking applied for writes. The register consists of 16 bits of data in [15:0], and 16 bits of corresponding masks in [31:16]. To set bit0, for example, the data would be 0x0001_0001. To clear bit0, for example, the data would be 0x0001_0000. Note that mask bit is the data bit offset + 16.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>RO</td>
</tr>
<tr>
<td>15:14</td>
<td></td>
<td>ECO Reserved 1</td>
</tr>
<tr>
<td></td>
<td>Format:</td>
<td>MBZ</td>
</tr>
<tr>
<td>13:8</td>
<td></td>
<td>ECO Reserved 2</td>
</tr>
<tr>
<td></td>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td></td>
<td>Format:</td>
<td>MBZ</td>
</tr>
<tr>
<td>7:6</td>
<td></td>
<td>ECO Reserved 3</td>
</tr>
<tr>
<td></td>
<td>Project:</td>
<td>All</td>
</tr>
<tr>
<td></td>
<td>Format:</td>
<td>MBZ</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>ECO_SCRATCH3B</td>
</tr>
<tr>
<td></td>
<td>Project:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Format:</td>
<td>MBZ</td>
</tr>
<tr>
<td>4:0</td>
<td></td>
<td>Sample_d Quality Mode</td>
</tr>
<tr>
<td></td>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td></td>
<td>Format:</td>
<td>U5</td>
</tr>
</tbody>
</table>

This field configures the image quality mode for the sample_d message in the sampling engine. In general, performance will increase with each step of reduced quality.

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
<th>Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>00h</td>
<td>Disabled</td>
<td>Full quality is enabled, matching prior products</td>
<td>All</td>
</tr>
<tr>
<td>01h-1Fh</td>
<td>Quality degrades with each larger value, performance improves with each larger value</td>
<td>All</td>
<td></td>
</tr>
</tbody>
</table>
**SAMPLER_READ_DATA**

<table>
<thead>
<tr>
<th>SAMPLER_RDATA - SAMPLER_READ_DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> RenderCS</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0xFFFFFFFF</td>
</tr>
<tr>
<td><strong>Access:</strong> RO Variant</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 0E144h</td>
</tr>
<tr>
<td><strong>Valid Projects:</strong> CHV, BSW</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FFFFFFFFh</td>
</tr>
</tbody>
</table>
# Save Timer

## SVTIMER - Save Timer

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x60001000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0B434h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td>30:29</td>
<td></td>
<td><strong>Counter Enabling Selection</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 11b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

LPFC provides rudimentary compression by allowing software to select from several predefined levels of event reporting. Based on the value of this bitfield, only a certain number of the programmed events in the "Event Selection and Base Counters" registers (CNT0CL, CNT1CL, ..., CNT7CL) will be tracked and reported:

<table>
<thead>
<tr>
<th>Value</th>
<th>Selected Counters</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>Counter 0</td>
</tr>
<tr>
<td>01</td>
<td>Counters 0 &amp; 1</td>
</tr>
<tr>
<td>10</td>
<td>Counters 0, 1, 2, &amp; 3</td>
</tr>
<tr>
<td>11</td>
<td>Counters 0 - 7</td>
</tr>
</tbody>
</table>

Signal - lpconf_lpf_cnt_enabled [1:0].

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>28:24</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td>23:0</td>
<td></td>
<td><strong>Save Timer Intervel</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000000100000000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

Save Timer Interval (SVTMRINT).

Save Timer Interval: This is the interval for sampling the performance counters and writing to memory. Each time it expires, the counters are sampled and packetized to be sent to DMA controller.

The minimum granularity of sampling period is 256 clocks. The value in this register is used as 256 x value to find the sampling window. For a 1Ghz core clock it provides up to 4ns of sampling period while matching the maximum capability of the event counters.

1h - 256clks.
2h - 512clks.
...
8h - 2048clks.

Signal - lpconf_lpf_savetimer_int [23:0].
## SB_ADDRESS

<table>
<thead>
<tr>
<th>SB_ADDRESS - SB_ADDRESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space:</td>
</tr>
<tr>
<td>Project:</td>
</tr>
<tr>
<td>Source:</td>
</tr>
<tr>
<td>Default Value:</td>
</tr>
<tr>
<td>Size (in bits):</td>
</tr>
<tr>
<td>Address:</td>
</tr>
</tbody>
</table>

The Sideband Address Register is used by the sideband transaction (triggered by Sideband Packet Register) for holding the address (of the internal register, within the destination unit).

IOSF SB access prevention: An IOSF SB write access targeting this register will complete with no affect. An IOSF SB read access targeting this register will abort (with 1's being returned on IOSF SB).

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>SB_Addr</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

Address, written by IOSF primary, for triggered IOSF SB initiated access.
0x18_2100 (SB_Busy) bit 0 = 1 then an IOSF primary write will NOT be captured.
0x18_2100 (SB_Busy) bit 0 = 0 then an IOSF primary write will be captured.
Dword aligned addresses must be used.
### SB_DATA - SB_DATA

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000h</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>182104h</td>
</tr>
</tbody>
</table>

The Sideband Data Register is used by the sideband register access mechanism (triggered by Sideband Packet Register) for holding write-data in write-transactions or read-data for read-transactions as explained below.

**IOSF SB access prevention:** An IOSF SB write access targeting this register will complete with no affect. An IOSF SB read access targeting this register will abort (with 1's being returned on IOSF SB).

**Requirement:** IOSF primary write to this register should not occur when a pending IOSF SB return is pending.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>SB Data</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Triggered IOSF SB write, this is the Data written in the IOSF SB initiated access.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Triggered IOSF SB read, this is the Data read return from the IOSF SB initiated read access.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0x18_2100 (SB_Busy) bit 0 = 1 then an IOSF primary write will NOT be captured.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0x18_2100 (SB_Busy) bit 0 = 0 then an IOSF primary write will be captured.</td>
</tr>
</tbody>
</table>
## SB_REQ_TRIGGER

<table>
<thead>
<tr>
<th>SB_REQ_TRIGGER - SB_REQ_TRIGGER</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong></td>
</tr>
<tr>
<td><strong>Project:</strong></td>
</tr>
<tr>
<td><strong>Source:</strong></td>
</tr>
<tr>
<td><strong>Default Value:</strong></td>
</tr>
<tr>
<td><strong>Size (in bits):</strong></td>
</tr>
<tr>
<td><strong>Address:</strong></td>
</tr>
</tbody>
</table>

When the Sideband Packet Register is written, the Gunit creates a transaction towards the destination agent on the IOSF sideband channel. The fields sent with the write operation as transaction parameters:

- The Sideband Rid field (bits 31:24) is used as the transaction Rid.
- The Sideband Opcode field (bits 23:15) is used as the transaction opcode.
- The Sideband Port field (bits 15:8) is used as the transaction destination port.
- The source port is hard-coded to 6h (Gunit port).
- The Sideband Byte Enable Field (bits 7:4) is used as the transaction Byte Enable.

If the opcode results in a data write semantic transaction, the write-data will be taken from the Sideband Data Register. If the opcode results in a data read semantic transaction, the read-data will be placed in the Sideband Data Register and may later be read by software. When Sideband Busy is set, Sideband Packet Register, Sideband Data Register and Sideband Address Register fields cannot be written. If the opcode results in a data read semantic transaction, data will be ready at Sideband Data register only when the Sideband Busy is cleared.

Fuse block access prevention: To prevent an attacker from using this mechanism to read the fuse block, writing the Fuse Block PortID to the Sideband Port will prevent the SB_busy bit from being set. This effectively 'aborts' the access. The triggered write is effectively dropped. The triggered read would supply whatever happened to be in the IOSF Sideband Doorbell Data register. IOSF SB access prevention: No usage models require IOSF SB accesses to the doorbell registers. No IOSF SB sources should be using IOSF SB accesses to the doorbell register to trigger a doorbell generated IOSF SB message. Software and firmware are PROHIBITED from using IOSF SB accesses to target and trigger accesses from this registers.

### Usage of the doorbell mechanism:
- **From IOSF primary, to initiate a write on IOSF Sideband:**
  a. Write Data (0x18_2104)
  b. Write Address (0x18_2108)
  c. Write Packet and Trigger (0x18_2100)
  d. Read poll 0x18_2100. When bit 0 = '0', then the write has completed on IOSF Sideband.
- **From IOSF primary, to initiate a read on IOSF Sideband:**
  a. Write Address (0x18_2108)
  b. Write Packet and Trigger (0x18_2100)
  c. Read poll 0x18_2100. When bit 0 = '0', then the read has completed on IOSF Sideband and data is available.
  d. Read IOSF SB returned Data (0x18_2104)

If software attempts a read to a register and the result is a UR (Unsupported Request), the busy bit will be cleared with no update of the doorbell data register.
### SB_REQ_TRIGGER - SB_REQ_TRIGGER

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>SB_DevFn</strong></td>
<td>00h</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Device and Function number to be used for the IOSF SB access. Per the IOSF Specification, this 8-bit field ('fid' in the IOSF Spec) is a unique identifier of a target in a hierarchy of PCI buses (indicates Device Number [31:27] / Function Number [26:24] or Function Number [31:24]). The target is free to utilize this in an agent-specific manner.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>23:16</td>
<td><strong>SB_Opcode</strong></td>
<td>00h</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Opcode to be used for the IOSF SB access.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>15:8</td>
<td><strong>SB_Port</strong></td>
<td>00h</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Port to be used for the IOSF SB access.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>7:4</td>
<td><strong>SB_ByteEnables</strong></td>
<td>0h</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Byte enables to be used for the IOSF SB access.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>3:1</td>
<td><strong>SB_BAR</strong></td>
<td>000b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BAR value</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td><strong>SB_Busy</strong></td>
<td>0b</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A write to this register will set this bit = '1' and triggers a IOSF SB request. When this bit is '1', the following registers are not IOSF primary write-able (0x18_2100, 0x18_2104, 0x18_2108) The completion of the IOSF SB access will clear this bit. 0 - 0x18_2100, 0x18_2104 and 0x18_2108 are writeable via IOSF Sideband. 1 - An IOSF SB access is in-progress. 0x18_2100, 0x18_2104 and 0x18_2108 are NOT writeable via IOSF Sideband.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## SCPD0 - SCPD0

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>18209Ch</td>
</tr>
</tbody>
</table>

### Scratch Pad 0 Register

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>Scratch Pad</td>
</tr>
</tbody>
</table>

*Default Value: 00000000h*

*Access: R/W*

Software scratch pad
## SCRATCH1 - SCRATCH1

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000 CHV, BSW</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0B11Ch</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>SCRATCH</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>
## SCRATCH for LNCFunit

<table>
<thead>
<tr>
<th>SCRATCH_LNCF1 - SCRATCH for LNCFunit</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000001 CHV, BSW</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 0B008h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:3</td>
<td>SCRATCH register for LNCFunit</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>Memory fill delay</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>lncf_csr_lni_gt2_memfill_dis.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0: mem fills gt2 latency will be 1.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1: mem fill gt2 latency will be same as gt3.</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>flush start delay</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>lncf_csr_lni_disable_flush_start_delay.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0: Flush processing in LNIunit starts one clock after receiving the flush command default.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1: Flush processing in LNI unit starts in the same clock in which flush command is received.</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td>Non-IA coherent atomics enable</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 1b</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0: atomics in GTI ()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1: atomics in L3 (non-IA atomic) (Default).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Output signal from LNCF unit lncf_csr_lni_glblatmcs_l3.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Value for this bit should be same as lbcf_csr_lsqc_glblatmcs_l3 b118[22].Value of this bit should be same as LBCF register bit 0xb11c[8].Adding Xbuf 8 MCP.</td>
</tr>
</tbody>
</table>

---

*Doc Ref # IHD-OS-CHV-BSW-Vol 2c- 10.15*
### Scratch Register 1

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space:</td>
<td>MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0A188h-0A18Bh</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Scratch1</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Register bits that have no connection to design. Used to enable/disable changes to the design that were put in during the ECO process.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>none</td>
</tr>
</tbody>
</table>
### Scratch Register 2

<table>
<thead>
<tr>
<th>SCRATCH2 - Scratch Register 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space:</td>
</tr>
<tr>
<td>Project:</td>
</tr>
<tr>
<td>Source:</td>
</tr>
<tr>
<td>Default Value:</td>
</tr>
<tr>
<td>Size (in bits):</td>
</tr>
<tr>
<td>Address:</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>Scratch2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>R/W</td>
</tr>
</tbody>
</table>

Register bits that have no connection to design. Used to enable/disable changes to the design that were put in during the ECO process.

none
# Second Buffer Size

## SBS - Second Buffer Size

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0B424h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td><strong>Second Virtual Buffer Base</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Second Virtual Buffer Base (SVBB0).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Second Virtual Buffer Base: Programmed by driver to allocate a memory space for performance data storage. The buffer size should be aligned to the size of the memory allocated so it naturally aligns to the base (i.e. for 128KB bit[16]=0, 256KB bit[17:16]=0, 512KB bit[18:16]=0).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Signal - lpconf_lpfc_virtual_base1 [31:16].</td>
</tr>
<tr>
<td>15:12</td>
<td></td>
<td><strong>Second Buffer Size 0</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td>11:0</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved.</td>
</tr>
</tbody>
</table>
## Second Level Batch Buffer Head Pointer Preemption Register

<table>
<thead>
<tr>
<th>SBB_PREEMPT_ADDR - Second Level Batch Buffer Head Pointer Preemption Register</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong></td>
</tr>
<tr>
<td><strong>Project:</strong></td>
</tr>
<tr>
<td><strong>Source:</strong></td>
</tr>
<tr>
<td><strong>Default Value:</strong></td>
</tr>
<tr>
<td><strong>Access:</strong></td>
</tr>
<tr>
<td><strong>Size (in bits):</strong></td>
</tr>
<tr>
<td><strong>Trusted Type:</strong></td>
</tr>
<tr>
<td><strong>Address:</strong></td>
</tr>
<tr>
<td><strong>Name:</strong></td>
</tr>
<tr>
<td><strong>ShortName:</strong></td>
</tr>
<tr>
<td><strong>Address:</strong></td>
</tr>
<tr>
<td><strong>Name:</strong></td>
</tr>
<tr>
<td><strong>ShortName:</strong></td>
</tr>
<tr>
<td><strong>Address:</strong></td>
</tr>
<tr>
<td><strong>Name:</strong></td>
</tr>
<tr>
<td><strong>ShortName:</strong></td>
</tr>
<tr>
<td><strong>Address:</strong></td>
</tr>
<tr>
<td><strong>Name:</strong></td>
</tr>
<tr>
<td><strong>ShortName:</strong></td>
</tr>
</tbody>
</table>

### Description

This register gets updated with the DWord-aligned graphics memory address of the PREEMPTABLE command in the second level batch buffer on which preemption has occurred.

This register value should be looked at only when the preemption has occurred in the second level batch buffer. This is indicated by "Ring/Batch Indicator" in "RING_BUFFER_HEAD_PREEMPT_REG". This register value retains its previous value and doesn't change when the preemption occurs on a preemptable command in ring buffer or in batch buffer.

Preemption is triggered by valid UHPTR in ring buffer mode of scheduling and by a pending execlist in Exec-List mode of scheduling. This is a global register and context save/restored as part of power context image.
# SBB_PREEMPT_ADDR - Second Level Batch Buffer Head Pointer

**Preemption Register**

<table>
<thead>
<tr>
<th>Preemptable Commands</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>MI_ARB_CHECK</td>
<td></td>
</tr>
<tr>
<td>3D_PRIMITIVE</td>
<td></td>
</tr>
<tr>
<td>GPGPU_WALKER</td>
<td></td>
</tr>
<tr>
<td>MEDIA_STATE_FLUSH</td>
<td></td>
</tr>
<tr>
<td>PIPE_CONTROL (Only in GPGPU mode of pipeline selection)</td>
<td></td>
</tr>
<tr>
<td>MI_ATOMIC (Post Sync Operation set in GPGPU mode of pipeline selection)</td>
<td></td>
</tr>
<tr>
<td>MI_SEMAPHORE_SIGNAL (Post Sync Operation set in GPGPU mode of pipeline selection)</td>
<td></td>
</tr>
</tbody>
</table>

**Programming Notes**

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:2</td>
<td><strong>Second Level Batch Buffer Head Pointer</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format:</strong> GraphicsAddress[31:2]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This field specifies the DWord-aligned Graphics Memory Address of the PREEMPTABLE command in a batch buffer where the Preemption has occurred.</td>
</tr>
<tr>
<td>1:0</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format:</strong> MBZ</td>
</tr>
</tbody>
</table>

Second Level Batch Buffer Head Pointer Register

<table>
<thead>
<tr>
<th>SBB_ADDR - Second Level Batch Buffer Head Pointer Register</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: CommandStreamer</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Access: RO</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 02114h-02117h</td>
</tr>
<tr>
<td>Name: Second Level Batch Buffer Head Pointer Register</td>
</tr>
<tr>
<td>ShortName: SBB_ADDR_RCSUNIT</td>
</tr>
<tr>
<td>Address: 12114h-12117h</td>
</tr>
<tr>
<td>Name: Second Level Batch Buffer Head Pointer Register</td>
</tr>
<tr>
<td>ShortName: SBB_ADDR_VCSUNIT0</td>
</tr>
<tr>
<td>Address: 1A114h-1A117h</td>
</tr>
<tr>
<td>Name: Second Level Batch Buffer Head Pointer Register</td>
</tr>
<tr>
<td>ShortName: SBB_ADDR_VECSUNIT</td>
</tr>
<tr>
<td>Address: 1C114h-1C117h</td>
</tr>
<tr>
<td>Name: Second Level Batch Buffer Head Pointer Register</td>
</tr>
<tr>
<td>ShortName: SBB_ADDR_VCSUNIT1</td>
</tr>
<tr>
<td>Address: 22114h-22117h</td>
</tr>
<tr>
<td>Name: Second Level Batch Buffer Head Pointer Register</td>
</tr>
<tr>
<td>ShortName: SBB_ADDR_BCSUNIT</td>
</tr>
</tbody>
</table>

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

Programming Notes

This register should NEVER be programmed by driver, this is for HW internal use only. This register should not be written by software. These fields should only get written by a context restore. Software should always set these fields via the MI_BATCH_BUFFER_START command when initiating a batch buffer. This register is saved and restored with context.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:2</td>
<td><strong>Second Level Batch Buffer Head Pointer</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: GraphicsAddress[31:2]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This field specifies the DWord-aligned Graphics Memory Address where the last initiated Second Level Batch Buffer is currently fetching commands. This field is meaningful only when Valid field is set to &quot;1&quot;.</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
</tbody>
</table>

Doc Ref # IHD-OS-CHV-BSW-Vol 2c-10.15

1163
## SBB_ADDR - Second Level Batch Buffer Head Pointer Register

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
<th>Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>Invalid</td>
<td>Second Level Batch buffer Invalid</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>1h</td>
<td>Valid</td>
<td>Second Batch buffer Valid.</td>
<td>CHV, BSW</td>
</tr>
</tbody>
</table>

**Valid**

Format: U1
## Second Level Batch Buffer State Register

<table>
<thead>
<tr>
<th>Register Name</th>
<th>Address Range</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SBB_STATE</td>
<td>02118h</td>
<td>This register contains the attributes of the second level batch buffer initiated from the batch Buffer. This register should not be written by software. These fields should only get written by a context restore. Software should always set these fields via the MI_BATCH_BUFFER_START command when initiating a batch buffer. This register is saved and restored with context.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:9</td>
<td>Reserved</td>
<td>MBZ</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>Reserved</td>
<td>CHV, BSW, MBZ</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>Resource Streamer Enable</td>
<td>U1</td>
</tr>
</tbody>
</table>

When this bit is set, the Resource Streamer will execute the batch buffer. When this bit is clear the Resource Streamer will not execute the batch buffer.
### SBB_STATE - Second Level Batch Buffer State Register

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td><strong>Reserved</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td></td>
<td>Format:</td>
<td>MBZ</td>
</tr>
<tr>
<td>5</td>
<td><strong>Address Space Indicator</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td></td>
<td>Note: 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.</td>
<td></td>
</tr>
<tr>
<td>0h</td>
<td>GGTT [Default]</td>
<td>This second level batch buffer is located in GGTT memory and is privileged</td>
</tr>
<tr>
<td>1h</td>
<td>PPGTT</td>
<td>This second level batch buffer is located in PPGTT memory and is non-privileged.</td>
</tr>
<tr>
<td>4</td>
<td><strong>Reserved</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td></td>
<td>Format:</td>
<td>MBZ</td>
</tr>
<tr>
<td>3:0</td>
<td><strong>Reserved</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Format:</td>
<td>MBZ</td>
</tr>
</tbody>
</table>
Second Level Batch Buffer Upper Head Pointer Preemption Register

<table>
<thead>
<tr>
<th><strong>SBB_PREEMPT_ADDR_UDW</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Access: R/W</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 02138h-0213Bh</td>
</tr>
<tr>
<td>Name: Second Level Batch Buffer Upper Head Pointer Preemption Register</td>
</tr>
<tr>
<td>ShortName: SBB_PREEMPT_ADDR_UDW_RCSUNIT</td>
</tr>
</tbody>
</table>

| Address: 12138h-1213Bh |
| Name: Second Level Batch Buffer Upper Head Pointer Preemption Register |
| ShortName: SBB_PREEMPT_ADDR_UDW_VCSUNIT0 |

| Address: 1A138h-1A13Bh |
| Name: Second Level Batch Buffer Upper Head Pointer Preemption Register |
| ShortName: SBB_PREEMPT_ADDR_UDW_VECSUNIT |

| Address: 1C138h-1C13Bh |
| Name: Second Level Batch Buffer Upper Head Pointer Preemption Register |
| ShortName: SBB_PREEMPT_ADDR_UDW_VCSUNIT1 |

| Address: 22138h-2213Bh |
| Name: Second Level Batch Buffer Upper Head Pointer Preemption Register |
| ShortName: SBB_PREEMPT_ADDR_UDW_BCSUNIT |

This register contains the 4GB aligned base address of gfx 4GB virtual address space within the host’s 64-bit virtual address space of the last preempted second level batch buffer. This register follows the same rules as the SBB_PREEMPT_ADDR register.

**Programming Notes**

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td>15:0</td>
<td><strong>Second Level Batch Buffer Head Pointer Upper DWORD</strong></td>
</tr>
</tbody>
</table>

**Format:** GraphicsAddress[47:32]

This field specifies the 4GB aligned base address of gfx 4GB virtual address space of the last preempted second level batch buffer.
## Second Level Batch Buffer Upper Head Pointer Register

**SBB_ADDR_UDW - Second Level Batch Buffer Upper Head Pointer Register**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>RO</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0211Ch-0211Fh</td>
</tr>
<tr>
<td>Name:</td>
<td>Second Level Batch Buffer Upper Head Pointer Register</td>
</tr>
<tr>
<td>ShortName:</td>
<td>SBB_ADDR_UDW_RCSUNIT</td>
</tr>
</tbody>
</table>

| Address:       | 1211Ch-1211Fh |
| Name:          | Second Level Batch Buffer Upper Head Pointer Register |
| ShortName:     | SBB_ADDR_UDW_VCSUNIT0 |

| Address:       | 1A11Ch-1A11Fh |
| Name:          | Second Level Batch Buffer Upper Head Pointer Register |
| ShortName:     | SBB_ADDR_UDW_VECSUNIT |

| Address:       | 1C11Ch-1C11Fh |
| Name:          | Second Level Batch Buffer Upper Head Pointer Register |
| ShortName:     | SBB_ADDR_UDW_VCSUNIT1 |

| Address:       | 2211Ch-2211Fh |
| Name:          | Second Level Batch Buffer Upper Head Pointer Register |
| ShortName:     | SBB_ADDR_UDW_BCSUNIT |

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

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td>Format:</td>
<td>MBZ</td>
</tr>
</tbody>
</table>

| 15:0 | Batch Buffer Head Pointer Upper DWORD |
|      | Format: GraphicsAddress[47:32] |

This field specifies the 4GB aligned base address of gfx 4GB virtual address space within the host’s 64-bit virtual address space where the last initiated Batch Buffer is currently fetching commands. If no batch buffer is currently active, the Valid bit in BB_ADDR will be 0 and this field is meaningless.
Semaphore Polling Interval on Wait

<table>
<thead>
<tr>
<th>SEMA_WAIT_POLL - Semaphore Polling Interval on Wait</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: RenderCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Access: R/W</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 0224Ch</td>
</tr>
<tr>
<td>Address: 1224Ch-1224Fh</td>
</tr>
<tr>
<td>Name: Semaphore Polling Interval on Wait</td>
</tr>
<tr>
<td>ShortName: SEMA_WAIT_POLL_VCSUNIT0</td>
</tr>
<tr>
<td>Address: 1A24Ch-1A24Fh</td>
</tr>
<tr>
<td>Name: Semaphore Polling Interval on Wait</td>
</tr>
<tr>
<td>ShortName: SEMA_WAIT_POLL_VECSUNIT</td>
</tr>
<tr>
<td>Address: 1C24Ch-1C24Fh</td>
</tr>
<tr>
<td>Name: Semaphore Polling Interval on Wait</td>
</tr>
<tr>
<td>ShortName: SEMA_WAIT_POLL_VCSUNIT1</td>
</tr>
<tr>
<td>Address: 2224Ch-2224Fh</td>
</tr>
<tr>
<td>Name: Semaphore Polling Interval on Wait</td>
</tr>
<tr>
<td>ShortName: SEMA_WAIT_POLL_BCSUNIT</td>
</tr>
</tbody>
</table>

The SEMA_WAIT_POLL register contains Poll Interval field which specifies the minimum number of microseconds allowed for command streamer to wait before re-fetching the data from the address mentioned in the MI_SEMAPHORE_WAIT command on WAIT Mode set to POLL until the condition is satisfied while the context is not switched out.

When a value of 0 is written the poll interval will be equal to the memory latency of the read completion.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:21</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
<tr>
<td></td>
<td>20:0</td>
<td><strong>Poll Interval</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Minimum number of micro-seconds allowed</td>
</tr>
</tbody>
</table>
### SSID_SID - SSID_SID

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>PCI: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0002Ch</td>
</tr>
</tbody>
</table>

This register is used to uniquely identify the subsystem where the PCI device resides.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td><strong>SUBID</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 0000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W Once</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This value is used to identify the vendor of the subsystem. This register is programmed by BIOS during boot-up. Once written, this register becomes Read Only. This register is cleared by a Reset.</td>
</tr>
<tr>
<td>15:0</td>
<td>SUBVID</td>
<td><strong>Default Value:</strong> 0000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W Once</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This value is used to identify the vendor of the subsystem. This register is programmed by BIOS during boot-up. Once written, this register becomes Read Only. This register is cleared by a Reset.</td>
</tr>
</tbody>
</table>
### GFXPAUSE - Staggered EU/SAMPLER PAUSE on Frequency Change

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0A000h-0A003h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:19</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>Pause Lock</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td>0 = Bits of EUPAUSE are R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 = All bits of EUPAUSE are RO (including this lock bit)</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>EU Pause Enable</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td>0 = Disabled; EUs will not be paused during frequency changes</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 = Enabled; EUs will be paused before graphics clocks are gated, and unpaused (staggered per EU) when the clocks are ungated</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Sampler Stall Enable</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td>0 = Disabled; Sampler will not be paused during frequency changes</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 = Enabled; Sampler will be paused before graphics clocks are gated, and unpaused when the clocks are ungated</td>
<td></td>
</tr>
<tr>
<td>15:0</td>
<td>Pause Count</td>
<td>R/W Lock</td>
</tr>
<tr>
<td></td>
<td>This is the minimum time the PMunit waits after asserting the EU or Sampler pause (if those are enabled) before allowing the core clocks to be gated.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0000 = Disabled</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0001 - Count 1 CSclk</td>
<td></td>
</tr>
<tr>
<td></td>
<td>...</td>
<td></td>
</tr>
<tr>
<td></td>
<td>FFFF = Count 65535 CSclks</td>
<td></td>
</tr>
</tbody>
</table>
## Storage 1

### STORAGE1 - Storage 1

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project: CHV, BSW</td>
<td></td>
</tr>
<tr>
<td>Source: PRM</td>
<td></td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0A500h-0A503h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td><strong>Wake Rate Counter Render</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>When reading, this field holds the number of times that the render well has awakened from RC1 (i.e. the number of times that (pmmr_cs_done &amp; ~gmcrgu_gpm_renderpower_req) changes from 1 to 0) within an evaluation interval.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>When writing this register, set the render wakerate counter to the value written.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The only reason to read this register is preparation for S0ix, though it may be useful for validation.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The only reason to write to this register is to resume from S0ix.</td>
</tr>
<tr>
<td>15:0</td>
<td></td>
<td><strong>Wake Rate Counter Media</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>When reading, this field holds the number of times that the media well has awakened from RC1 (i.e. the number of times that (pmmr_vcs_done &amp; pmmr_bcs_done &amp; ~gmcrgu_gpm_mediapower_req) changes from 1 to 0) within an evaluation interval.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>When writing this register, set the render wakerate counter to the value written.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The only reason to read this register is preparation for S0ix, though it may be useful for validation.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The only reason to write to this register is to resume from S0ix.</td>
</tr>
</tbody>
</table>
## Storage 2

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0A504h-0A507h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td>Reserved</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access:</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>23:0</td>
<td>RC El Counter Media</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access:</td>
<td></td>
</tr>
</tbody>
</table>
### STORAGE3 - Storage 3

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
</tbody>
</table>

**Address:** 0A508h-0A50Bh

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>Reserved</strong></td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td>23:0</td>
<td><strong>RC EI Counter Render</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## STORAGE4 - Storage 4

<table>
<thead>
<tr>
<th>Address</th>
<th>0A50Ch-0A50Fh</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DWord</strong></td>
<td><strong>Bit</strong></td>
</tr>
<tr>
<td>0</td>
<td>31:24</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>23:0</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### STORAGE5 - Storage 5

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0A510h-0A513h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Access:</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td>Reserved</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td>23:0</td>
<td><strong>RC Idle Counter Render</strong></td>
<td>R/W</td>
</tr>
</tbody>
</table>
# Storage 6

<table>
<thead>
<tr>
<th>STORAG6 - Storage 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 0A514h-0A517h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>0</td>
<td>23:0</td>
<td>RP EI Up Counter</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>
## STORAGE7 - Storage 7

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>Reserved</strong></td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td>23:0</td>
<td><strong>RP EI Up Busy Counter</strong></td>
<td>R/W</td>
</tr>
</tbody>
</table>

**Register Space:** MMIO: 0/2/0  
**Project:** CHV, BSW  
**Source:** PRM  
**Default Value:** 0x00000000  
**Size (in bits):** 32  
**Address:** 0A518h-0A51Bh
## STORAGE8 - Storage 8

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
<th>Project:</th>
<th>CHV, BSW</th>
<th>Source:</th>
<th>PRM</th>
<th>Default Value:</th>
<th>0x00000000</th>
<th>Size (in bits):</th>
<th>32</th>
<th>Address:</th>
<th>0A51Ch-0A51Fh</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Access:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td>Reserved</td>
<td>RO</td>
<td></td>
</tr>
<tr>
<td>23:0</td>
<td></td>
<td>RP EI Down Counter</td>
<td>R/W</td>
<td></td>
</tr>
</tbody>
</table>
## STORAGE9 - Storage 9

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0A520h-0A523h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td>Reserved</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23:0</td>
<td></td>
<td>RP EI Down Busy Counter</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Stream Output Num Primitives Written Counter

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

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>RenderCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000, 0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>64</td>
</tr>
<tr>
<td>Address:</td>
<td>05200h-0521Fh</td>
</tr>
</tbody>
</table>

There is one 64-bit register for each of the 4 supported streams:
- SO_NUM_PRIMS_WRITTEN0 (for Stream Out Stream #0)
- SO_NUM_PRIMS_WRITTEN1 (for Stream Out Stream #1)
- SO_NUM_PRIMS_WRITTEN2 (for Stream Out Stream #2)
- SO_NUM_PRIMS_WRITTEN3 (for Stream Out Stream #3)

These registers are used to count the number of primitives (aka objects: points, lines, triangles) which the SO stage has successfully written to a particular "stream's" Streamed Vertex Output buffers, subject to buffer overflow detection. (See the Stream Output section of the 3D pipeline volume). These registers are part of the context save and restore.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>63:0</td>
<td><strong>Num Prims Written Count</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format:</strong> U64</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This count is incremented (by one) every time a GS thread outputs a DataPort Streamed Vertex Buffer Write message with the Increment Num Prims Written bit set in the message header (see the Geometry Shader and Data Port chapters in the 3D Volume.)</td>
</tr>
</tbody>
</table>
Stream Output Primitive Storage Needed Counters

<table>
<thead>
<tr>
<th>SO_PRIM_STORAGE_NEEDED[0:3] - Stream Output Primitive Storage Needed Counters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: RenderCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000, 0x00000000</td>
</tr>
<tr>
<td>Access: RW. This register is set by the context restore.</td>
</tr>
<tr>
<td>Size (in bits): 64</td>
</tr>
</tbody>
</table>

Address: 05240h-0525Fh

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

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

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

These registers are part of the context save and restore.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>63:0</td>
<td><strong>Prim Storage Needed Count</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: U64</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This count is incremented (by one) by the SOL stage for each object (point, line, triangle) it writes or attempts to write to the corresponding stream's output buffers. The count is not affected by the actual number of buffers bound to the stream.</td>
</tr>
</tbody>
</table>
Stream Output Write Offsets

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>RenderCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>RW. This register is set by the context restore.</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>05280h-0528Fh</td>
</tr>
</tbody>
</table>

There is one R/W 32-bit register for each of the 4 supported stream output buffer slots:
- 5280h-5283h SO_WRITE_OFFSET0 (for Stream Out Buffer #0)
- 5284h-5287h SO_WRITE_OFFSET1 (for Stream Out Buffer #1)
- 5288h-528Bh SO_WRITE_OFFSET2 (for Stream Out Buffer #2)
- 528Ch-528Fh SO_WRITE_OFFSET3 (for Stream Out Buffer #3)

These registers are used to set and track a DWord-granular Write Offset for each of the 4 Stream Output Buffer slots. Software can directly write them via MI_LOAD_REGxxx commands. The SOL stage will increment them as part of stream output processing. Software can cause them to be written to memory via MI_STORE_REGxxx commands. (See the Stream Output section of the 3D pipeline volume).

These registers are part of the context save and restore.

### Programming Notes

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:2</td>
<td><strong>Write Offset</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Project</strong>: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong>: U30</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This field contains a DWord offset from the corresponding SO buffer’s Base Address value. The SOL stage uses this value as a write offset when performing writes to the buffer. The SOL stage will increment this value as a part of performing stream output to the buffer. Note that the SOL stage uses the buffer’s Surface Pitch to advance the Write Offset, without regard to the buffer’s Base Address.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>1:0</th>
<th><strong>Reserved</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong>: MBZ</td>
</tr>
</tbody>
</table>
## SWF1

### SWF1 - SWF1

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>4F000h</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these register is defined by software architecture.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>SOFTWARE_FLAGS</strong></td>
</tr>
</tbody>
</table>

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Value:</td>
<td>00000000h</td>
<td></td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
<td></td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these register is defined by software architecture.
### SWF2 - SWF2

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>4F004h</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these register is defined by software architecture.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>SOFTWARE_FLAGS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these register is defined by software architecture.
### SWF3 - SWF3

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>4F008h</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these register is defined by software architecture.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>SOFTWARE_FLAGS</td>
</tr>
</tbody>
</table>

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Value:</td>
<td>00000000h</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these register is defined by software architecture.
## SWF4 - SWF4

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>4F00Ch</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these registers is defined by software architecture.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>SOFTWARE_FLAGS</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these registers is defined by software architecture.
**SWF5 - SWF5**

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>SOFTWARE_FLAGS</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these registers is defined by software architecture.
### SWF6 - SWF6

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>4F014h</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these register is defined by software architecture.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>SOFTWARE_FLAGS</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these register is defined by software architecture.
### SWF7 - SWF7

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>4F018h</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these register is defined by software architecture.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>SOFTWARE_FLAGS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these register is defined by software architecture.
## SWF8

### SWF8 - SWF8

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>4F01Ch</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these register is defined by software architecture.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>SOFTWARE_FLAGS</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these register is defined by software architecture.
## SWF9 - SWF9

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>SOFTWARE_FLAGS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these registers is defined by software architecture.
### SWF10 - SWF10

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>4F024h</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these register is defined by software architecture.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>SOFTWARE_FLAGS</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these register is defined by software architecture.
## SWF11 - SWF11

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>4F028h</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these register is defined by software architecture.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>SOFTWARE_FLAGS</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>R/W</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these register is defined by software architecture.
## SWF12 - SWF12

<table>
<thead>
<tr>
<th><strong>Register Space:</strong></th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Project:</strong></td>
<td>CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong></td>
<td>PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong></td>
<td>0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong></td>
<td>32</td>
</tr>
<tr>
<td><strong>Address:</strong></td>
<td>4F02Ch</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these register is defined by software architecture.

<table>
<thead>
<tr>
<th><strong>DWord</strong></th>
<th><strong>Bit</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>SOFTWARE_FLAGS</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these register is defined by software architecture.
**SWF13 - SWF13**

Register Space: MMIO: 0/2/0  
Project: CHV, BSW  
Source: PRM  
Default Value: 0x00000000  
Size (in bits): 32  
Address: 4F030h

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these registers is defined by software architecture.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>SOFTWARE_FLAGS</strong></td>
</tr>
</tbody>
</table>

Default Value: 00000000h  
Access: R/W

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these registers is defined by software architecture.
## SWF14 - SWF14

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>4F034h</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these register is defined by software architecture.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>SOFTWARE_FLAGS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these register is defined by software architecture.
**Register Space:** MMIO: 0/2/0  
**Project:** CHV, BSW  
**Source:** PRM  
**Default Value:** 0x00000000  
**Size (in bits):** 32

**Address:** 4F038h

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these register is defined by software architecture.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>SOFTWARE_FLAGS</td>
</tr>
</tbody>
</table>

| Default Value: | 00000000h |
| Access:        | R/W       |

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these register is defined by software architecture.
### SWF16 - SWF16

<table>
<thead>
<tr>
<th>Register Space</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits)</td>
<td>32</td>
</tr>
<tr>
<td>Address</td>
<td>4F03Ch</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these register is defined by software architecture.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>SOFTWARE_FLAGS</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these register is defined by software architecture.
### SWF17 - SWF17

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>4F040h</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these register is defined by software architecture.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>SOFTWARE_FLAGS</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Default Value:</td>
</tr>
<tr>
<td></td>
<td>Access:</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these register is defined by software architecture.
## SWF18

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>SOFTWARE_FLAGS</strong></td>
</tr>
</tbody>
</table>

- **Default Value:** 00000000h
- **Access:** R/W

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these registers is defined by software architecture.
### SWF19 - SWF19

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>4F048h</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these registers is defined by software architecture.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>SOFTWARE_FLAGS</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these registers is defined by software architecture.
### SWF20 - SWF20

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>4F04Ch</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these register is defined by software architecture.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>SOFTWARE_FLAGS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these register is defined by software architecture.
**SWF21 - SWF21**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>4F050h</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these registers is defined by software architecture.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>SOFTWARE_FLAGS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these registers is defined by software architecture.
## SWF22

<table>
<thead>
<tr>
<th>SWF22 - SWF22</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 4F054h</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these registers is defined by software architecture.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>SOFTWARE_FLAGS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these registers is defined by software architecture.
### SWF23 - SWF23

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>4F058h</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these registers is defined by software architecture.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>SOFTWARE_FLAGS</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these registers is defined by software architecture.
## SWF24

<table>
<thead>
<tr>
<th>SWF24 - SWF24</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 4F05Ch</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these register is defined by software architecture.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>SOFTWARE_FLAGS</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these register is defined by software architecture.
## SWF25 - SWF25

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>4F060h</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these register is defined by software architecture.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>SOFTWARE_FLAGS</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these register is defined by software architecture.
### SWF26 - SWF26

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>4F064h</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these registers is defined by software architecture.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>SOFTWARE_FLAGS</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these registers is defined by software architecture.
## SWF27 - SWF27

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>SOFTWARE_FLAGS</td>
</tr>
</tbody>
</table>

- **Default Value:** 00000000h
- **Access:** R/W

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation.
The use of these register is defined by software architecture.

Register Space: MMIO: 0/2/0  
Project: CHV, BSW  
Source: PRM  
Default Value: 0x00000000  
Size (in bits): 32  
Address: 4F068h

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation.
The use of these register is defined by software architecture.
**SWF28 - SWF28**

Register Space: MMIO: 0/2/0  
Project: CHV, BSW  
Source: PRM  
Default Value: 0x00000000  
Size (in bits): 32  
Address: 4F06Ch

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these registers is defined by software architecture.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>SOFTWARE_FLAGS</strong></td>
<td>00000000h</td>
<td>R/W</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these registers is defined by software architecture.
## SWF29 - SWF29

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>4F070h</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these registers is defined by software architecture.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>SOFTWARE_FLAGS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these registers is defined by software architecture.
## SWF30

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>4F074h</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these register is defined by software architecture.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>SOFTWARE_FLAGS</td>
<td>00000000h</td>
<td>R/W</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these register is defined by software architecture.
**SWF31 - SWF31**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>4F078h</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these register is defined by software architecture.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>SOFTWARE_FLAGS</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these register is defined by software architecture.
## SWF32

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>4F07Ch</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these register is defined by software architecture.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>SOFTWARE_FLAGS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these register is defined by software architecture.
### SWF33 - SWF33

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>4F080h</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these register is defined by software architecture.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>SOFTWARE_FLAGS</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these register is defined by software architecture.
## SWF34

### SWF34 - SWF34

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>4F084h</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these registers is defined by software architecture.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>SOFTWARE_FLAGS</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Default Value:</th>
<th>00000000h</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these registers is defined by software architecture.
### SWF35 - SWF35

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>4F088h</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these register is defined by software architecture.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>SOFTWARE_FLAGS</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access:</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these register is defined by software architecture.
### SWF36

<table>
<thead>
<tr>
<th>SWF36 - SWF36</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong></td>
</tr>
<tr>
<td><strong>Project:</strong></td>
</tr>
<tr>
<td><strong>Source:</strong></td>
</tr>
<tr>
<td><strong>Default Value:</strong></td>
</tr>
<tr>
<td><strong>Size (in bits):</strong></td>
</tr>
<tr>
<td><strong>Address:</strong></td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these register is defined by software architecture.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>SOFTWARE_FLAGS</td>
</tr>
<tr>
<td></td>
<td>Default Value:</td>
<td>00000000h</td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>R/W</td>
</tr>
</tbody>
</table>

These (36 DW) registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these register is defined by software architecture.
**SWSMISCI - SWSMISCI**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>PCI: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>000E0h</td>
</tr>
</tbody>
</table>

Software SMI or SCI.
The SCI mechanism for driver / BIOS communication. SMI is a system wide lock interrupt (halts all the cores) as opposed to SCI.

The SMI is slowly being phased out.
This register serves 2 purposes: 1) Support selection of SMI or SCI event source (SMISCISEL - bit15) 2) Event trigger (bit 0).

To generate a SW SCI event, software (System BIOS/Graphics driver) should program bit 15 (SMISCISEL) to 1. This is typically programmed once (assuming SMIs are never triggered). On a '0' to '1' subsequent transition in bit 0 of this register (caused by a software write operation), GMCH sends a single SCI message. The SCI will set the DMISCI bit in its TCO1_STS register and TCOSCI_STS bit in its GPE0 register upon receiving this message from DMI.

Once written as 1, software must write a '0' to this bit to clear it, and all other write transitions (1 ->0, 0 ->0, 1->1) or if bit 15 is '0' will not cause GMCH to send SCI message to DMI link.

To generate a SW SMI event, software should program bit 15:0 and trigger SMI.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td>RESERVED</td>
</tr>
<tr>
<td>15</td>
<td>SMI OR SCI_EVENT_SELECT</td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MCS: SMI or SCI event select. 0 = SMI, 1 = SCI</td>
</tr>
<tr>
<td>14:1</td>
<td>SOFTWARE_SCRATCH_BITS</td>
<td>Default Value: 0000h</td>
</tr>
<tr>
<td></td>
<td>Used by driver to communicate information to SBIOS</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>SMI OR SCI_EVENT</td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td>MCE: MCS=1, setting this bit causes an SCI. MCS=0, setting this bit causes an SMI. A 1 to 0, 0 to 0 or 1 to 1 transition of this bit does not trigger any events. The graphics driver writes to this register as a means to interrupt the SBIOS.</td>
<td></td>
</tr>
</tbody>
</table>
Thread Dispatched Count Register

**TDL_THR_DISP_COUNT - Thread Dispatched Count Register**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>RO</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0E4BCh</td>
</tr>
<tr>
<td>Valid Projects:</td>
<td>CHV, BSW</td>
</tr>
</tbody>
</table>

This register provides the count of threads dispatched/valid in the subslice.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:6</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
<tr>
<td>5:0</td>
<td></td>
<td>Thread Count</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-56</td>
<td>Valid Range</td>
</tr>
</tbody>
</table>
## Thread Faulted Count Register

**TDL_THR_PF_COUNT - Thread Faulted Count Register**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>RO</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0E5BCh</td>
</tr>
<tr>
<td>Valid Projects:</td>
<td>CHV, BSW</td>
</tr>
</tbody>
</table>

This register provides the count of threads faulted in each subslice.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31</td>
<td><strong>Canonical fault indication bit to CS</strong>&lt;br&gt;The bit is set when a canonical fault on data fetch is reported by EU.</td>
</tr>
<tr>
<td>30:6</td>
<td></td>
<td><strong>Reserved</strong>&lt;br&gt;Format: MBZ</td>
</tr>
<tr>
<td>5:0</td>
<td></td>
<td><strong>Thread Count</strong>&lt;br&gt;Value</td>
</tr>
<tr>
<td></td>
<td>0-56</td>
<td>Valid Range</td>
</tr>
</tbody>
</table>

1222

Doc Ref # IHD-OS-CHV-BSW-Vol 2c-10.15
Thread Fault Status Register 0

<table>
<thead>
<tr>
<th>TDL_THR_PF_STATUS0 - Thread Fault Status Register 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Access: RO</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 0E6B8h</td>
</tr>
</tbody>
</table>

This register provides the status of each thread in the SubSlice. A bit set indicates the thread in the specific thread slot is faulted.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td>Row0, EU3, [Reserved, T6-T0]</td>
</tr>
<tr>
<td></td>
<td>23:16</td>
<td>Row0, EU2, [Reserved, T6-T0]</td>
</tr>
<tr>
<td></td>
<td>15:8</td>
<td>Row0, EU1, [Reserved, T6-T0]</td>
</tr>
<tr>
<td></td>
<td>7:0</td>
<td>Row0, EU0, [Reserved, T6-T0]</td>
</tr>
</tbody>
</table>
Thread Fault Status Register 1

<table>
<thead>
<tr>
<th>TDL_THR_PF_STATUS1 - Thread Fault Status Register 1</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 0E7B8h</td>
</tr>
</tbody>
</table>

This register provides the status of each thread in the SubSlice. A bit set indicates the thread in the specific thread slot is faulted.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td>Row1, EU3, [Reserved, T6-T0]</td>
</tr>
<tr>
<td></td>
<td>23:16</td>
<td>Row1, EU2, [Reserved, T6-T0]</td>
</tr>
<tr>
<td></td>
<td>15:8</td>
<td>Row1, EU1, [Reserved, T6-T0]</td>
</tr>
<tr>
<td></td>
<td>7:0</td>
<td>Row1, EU0, [Reserved, T6-T0]</td>
</tr>
</tbody>
</table>
Thread Load Status Register 0

TDL_THR_STATUS0 - Thread Load Status Register 0

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>RO</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0E4B8h</td>
</tr>
</tbody>
</table>

This register provides the status of each thread in the SubSlice.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td>Row0, EU3, [Reserved, T6-T0]</td>
</tr>
<tr>
<td></td>
<td>23:16</td>
<td>Row0, EU2, [Reserved, T6-T0]</td>
</tr>
<tr>
<td></td>
<td>15:8</td>
<td>Row0, EU1, [Reserved, T6-T0]</td>
</tr>
<tr>
<td></td>
<td>7:0</td>
<td>Row0, EU0, [Reserved, T6-T0]</td>
</tr>
</tbody>
</table>
Thread Load Status Register 1

**TDL_THR_STATUS1 - Thread Load Status Register 1**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>RO</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0E5B8h</td>
</tr>
</tbody>
</table>

This register provides the status of each thread in the SubSlice. A bit set indicates a valid thread is loaded in the thread slot.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td>Row1, EU3, [Reserved, T6-T0]</td>
</tr>
<tr>
<td></td>
<td>23:16</td>
<td>Row1, EU2, [Reserved, T6-T0]</td>
</tr>
<tr>
<td></td>
<td>15:8</td>
<td>Row1, EU1, [Reserved, T6-T0]</td>
</tr>
<tr>
<td></td>
<td>7:0</td>
<td>Row1, EU0, [Reserved, T6-T0]</td>
</tr>
</tbody>
</table>
Thread Mode Register

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
<tr>
<td>30</td>
<td></td>
<td>TDS external Cache Disable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0b</td>
<td>Enable</td>
<td>The external TDS Cache is enabled if there is enough handles to enable the cache.</td>
</tr>
<tr>
<td>1b</td>
<td>Disable</td>
<td>The external TDS Cache is disabled even if there is enough handles to enable the cache. Only the internal TDS Cache will be used.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>[Default]</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>[1,9]</td>
<td></td>
<td>CHV, BSW</td>
</tr>
</tbody>
</table>

Valid Projects:

This register is used to program the FF shader Mode.
## FF_MODE - Thread Mode Register

### 25:20 VS Hit Max Value

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>[Default]</td>
<td>If the number of hits reaches the VS Hit Max Value and there is a pending miss to be dispatched, the VS will dispatch the pending miss vertex as a single dispatch. Programming the value beyond the range will have undefined behavior if VS Reference Count Full Force miss enable is 0. When VS Reference Count Full Force miss enable is 1 then the value can be [1,3Fh].</td>
</tr>
<tr>
<td>[1,26]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 19 DS Reference Count Full Force Miss Enable

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0b</td>
<td>[Default]</td>
<td>On a hit to the DS cache and the associated handle's reference count is full then stall until a dereference.</td>
</tr>
<tr>
<td>1b</td>
<td></td>
<td>On a hit to the DS cache and the associated handle's reference count is full then force the cycle as a miss and allocate a new handle.</td>
</tr>
</tbody>
</table>

**Programming Notes**

To work around bugs, this must be set to 0.

### 18 Reserved

### 17:16 Reserved

### 15 VS Reference Count Full Force Miss Enable

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0b</td>
<td>[Default]</td>
<td>On a hit to the VS cache and the associated handle's reference count is full then stall until a dereference.</td>
</tr>
<tr>
<td>1b</td>
<td></td>
<td>On a hit to the VS cache and the associated handle's reference count is full then force the cycle as a miss and allocate a new handle.</td>
</tr>
</tbody>
</table>

**Programming Notes**

To work around bugs, this must be set to 0.
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Default Value</th>
<th>Project</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>14:13</td>
<td>Reserved</td>
<td></td>
<td>CHV, BSW</td>
<td>MBZ</td>
</tr>
<tr>
<td>12</td>
<td>Reserved</td>
<td>0h</td>
<td>CHV, BSW</td>
<td>MBZ</td>
</tr>
<tr>
<td>11:7</td>
<td>Reserved</td>
<td></td>
<td>CHV, BSW</td>
<td>MBZ</td>
</tr>
<tr>
<td>6:5</td>
<td>Reserved</td>
<td></td>
<td>CHV, BSW</td>
<td>MBZ</td>
</tr>
<tr>
<td>4</td>
<td>Reserved</td>
<td>0h</td>
<td>CHV, BSW</td>
<td>MBZ</td>
</tr>
<tr>
<td>3:0</td>
<td>Reserved</td>
<td></td>
<td>CHV, BSW</td>
<td>MBZ</td>
</tr>
</tbody>
</table>
Thread Restart Control Register

**TDL_THR_RESTART - Thread Restart Control Register**

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:1</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td><strong>Restart All Faulted Threads</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>A write of 1 to this register restarts all threads that have halted due to page fault.</td>
</tr>
</tbody>
</table>
## TILECTL

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>101000h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:3</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td><strong>TLBPFL</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Store multiple PTE enable</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0: Only one Page Table Entry</td>
</tr>
<tr>
<td></td>
<td></td>
<td>is stored in the Translation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Lookaside Buffer cache for</td>
</tr>
<tr>
<td></td>
<td></td>
<td>tiled cycles.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1: Multiple Page Table Entries</td>
</tr>
<tr>
<td></td>
<td></td>
<td>are stored in the Translation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Lookaside Buffer cache for</td>
</tr>
<tr>
<td></td>
<td></td>
<td>tiled cycles.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If tileX, then 4 entries are</td>
</tr>
<tr>
<td></td>
<td></td>
<td>stored. If tileY, then 8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>entries are stored.</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td><strong>SWZCTL</strong></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Not used for CHV, BSW.</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>This register location is</td>
</tr>
<tr>
<td></td>
<td></td>
<td>updated via GFX Driver prior</td>
</tr>
<tr>
<td></td>
<td></td>
<td>to enabling DRAM accesses.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The Driver needs to obtain the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>need for memory address</td>
</tr>
<tr>
<td></td>
<td></td>
<td>swizzling via DRAM configuration</td>
</tr>
<tr>
<td></td>
<td></td>
<td>registers and set the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>following bits.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>x0b - No Address Swizzling</td>
</tr>
<tr>
<td></td>
<td></td>
<td>x1b - Address bit [6] needs to</td>
</tr>
<tr>
<td></td>
<td></td>
<td>be swizzled for tiled surfaces</td>
</tr>
</tbody>
</table>
## TiledResources Invalid Tile Detection Register

<table>
<thead>
<tr>
<th>TRINVTILEDETCT - TiledResources Invalid Tile Detection Register</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 04DECh</td>
</tr>
<tr>
<td>Name: TiledResources Invalid Tile Detection Register</td>
</tr>
<tr>
<td>ShortName: TRINVTILEDETCT</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Invalid Tile Detection Value</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000000h</td>
<td>[Default]</td>
<td>A 32bit value programmed to enable h/w to perform a match of TR-VA TT entries to detect Invalid Tiles. Hardware will flag each entry and space behind it as Invalid Tile for matched entries.</td>
</tr>
</tbody>
</table>
# Tiled Resources Translation Table Control Registers

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:2</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> RO</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000000000000000000000000b</td>
<td>[Default]</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td><strong>TR-VA Translation Table Memory Location</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
</tbody>
</table>

This field specifies whether the translation tables for TR to VA are in virtual address space v/s physical (GPA) address space.
0: Tables are in Physical (GPA) space
1: Tables are in Virtual address space

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td><strong>TR - TT Enable</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
</tbody>
</table>

TR translation tables are disabled as default.
This field needs to be enabled via s/w to get TR translation active.
# TiledResources VA Detection Registers

<table>
<thead>
<tr>
<th><strong>TRVADR - TiledResources VA Detection Registers</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 04DF0h</td>
</tr>
<tr>
<td>Name: TiledResources VA Detection Registers</td>
</tr>
<tr>
<td>ShortName: TRVADR</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:8</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>7:4</td>
<td>TR</td>
<td>VA Mask Value</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4bit MASK value that is mapped to incoming address bits[47:44]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MASK bits are used to identify which address bits need to be considered for compare.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If particular mask bit is &quot;1&quot;, mapping address bit needs to be compared to DATA value provided.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If &quot;0&quot;, corresponding address bit is masked which makes it don't care for compare. (This field defaults to &quot;0000&quot; to disable detection.).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note: The only usage model for GFX driver to set this field to &quot;1111&quot;. Behaviour of h/w for any other setting is not defined.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note: GFX driver shall use same TRVA MASK value for all contexts.</td>
</tr>
<tr>
<td>3:0</td>
<td>TR</td>
<td>VA Data Value</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000b</td>
<td>[Default]</td>
<td>4bit Data value that is mapped to incoming address bits[47:44]. Data bits are used to compare address values that are not filtered by the TRVAMV for match.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note: GFX driver shall use same TRVA Data value for all contexts</td>
</tr>
</tbody>
</table>
## Tiled Resources VA Translation Table L3 ptr - DW0

**TRVATTL3PTRDW0 - Tiled Resources VA Translation Table L3 ptr - DW0**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Address:</th>
<th>04DE0h</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name:</td>
<td>Tiled Resources VA Translation Table L3 ptr - DW0</td>
</tr>
<tr>
<td>ShortName:</td>
<td>TRVATTL3PTRDW0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:12</td>
<td>TR - VA transIn Table L3 Pointer (Lower Address)</td>
<td>00000h</td>
<td>[Default]</td>
<td>Lower address bits for tiled resource VA to virtual address translation L3 table</td>
</tr>
<tr>
<td>11:0</td>
<td></td>
<td>Reserved</td>
<td></td>
<td></td>
<td>000h</td>
</tr>
</tbody>
</table>

**Access:**
- **R/W**
- **RO**

**Default Value:**
- 000h

**Access:**
- **RO**

**Reserved**
### TRVATTL3PTRDW1 - Tiled Resources VA Translation Table L3 ptr - DW1

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Value</th>
<th>Name</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td><strong>Reserved</strong></td>
<td>0000h</td>
<td>[Default]</td>
<td>RO</td>
<td></td>
</tr>
<tr>
<td></td>
<td>15:0</td>
<td><strong>TR - VA transIn Table L3 Pointer (Upper Address)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0000h</td>
<td></td>
<td></td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Upper address bits for tiled resource VA to virtual address translation L3 table</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Register Space: MMIO: 0/2/0  
Source: PRM  
Default Value: 0x00000000  
Size (in bits): 32  
Address: 04DE4h  
Name: Tiled Resources VA Translation Table L3 ptr - DW1  
ShortName: TRVATTL3PTRDW1
## TLB_RD_ADDRESS Register

<table>
<thead>
<tr>
<th>TLB_RD_ADDR - TLB_RD_ADDRESS Register</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 04B00h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:12</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 000000000000000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>11:0</td>
<td></td>
<td>Reserved</td>
</tr>
</tbody>
</table>
## TLB_RD_DATA0 Register

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>04B04h</td>
</tr>
</tbody>
</table>

### TLBReadWriteData0 Register

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>TLBReadWriteData0 Register</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO address [43:12]</td>
</tr>
</tbody>
</table>
## TLB_RD_DATA1 Register

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
</tbody>
</table>

**Address:** 04B08h

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>TLB_READ_DATA1 Register</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[31:5] Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit[3:0] address [47:44]</td>
</tr>
</tbody>
</table>
## Transcode Attack Status Register

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:1</td>
<td><strong>RSVD</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>RSVD</strong></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td><strong>GKEYS_STATUS</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Transcode Attack Status bit</td>
</tr>
</tbody>
</table>
### TRNULLDETCT

| Register Space: | MMIO: 0/2/0 |
| Source: | PRM |
| Default Value: | 0x00000000 |
| Size (in bits): | 32 |
| Address: | 04DE8h |
| Name: | TiledResources Null Tile Detection Register |
| ShortName: | TRNULLDETCT |

#### DWord | Bit | Description
--- | --- | ---
0 | 31:0 | **Null Tile Detection Value**

**Access:** R/W

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000000h</td>
<td>[Default]</td>
<td>A 32bit value programmed to enable h/w to perform a match of TR-VA TT entries to detect Null Tiles. Hardware will flag each entry and space behind it as Null Tile for matched entries.</td>
</tr>
</tbody>
</table>
# Turbo Media Control

## TMCTL - Turbo Media Control

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0A190h-0A193h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:9</td>
<td><strong>Reserved</strong></td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td><strong>Bypass Media Idle Hysteresis Enable (aka Slice Shutdown)</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>7:1</td>
<td>Reserved</td>
<td>RO</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td><strong>Bypass Render Idle Hysteresis Enable (aka Slice Shutdown)</strong></td>
<td>R/W</td>
</tr>
</tbody>
</table>

1 - Bypass idle hysteresis requirements for making an RC wish.
0 - Honor idle hysteresis <default>
Unblock Message Act to Busy Detection Timer

<table>
<thead>
<tr>
<th>RCUBMABDTMR - Unblock Message Act to Busy Detection Timer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 0A0B0h-0A0B3h</td>
</tr>
</tbody>
</table>

When locked exit policy is chosen, the GT core can be woken up in parallel with the IA cores waking up so an MMIO write to GT core might not be coming, or may be later in coming. If after the time programmed in this register is met and graphics is still idle, then RC1(e)/RC6x is allowed to be entered.

0 = 0 usec
1 = 1.28 usec
2 = 2.56 usec
3 = 3.84 usec
FF FFFF = 21.474 sec

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td>23:0</td>
<td></td>
<td>Unblock Message Ack to Busy Detection Timer</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

When locked exit policy is chosen, the GT core can be woken up in parallel with the IA cores waking up so an MMIO write to GT core might not be coming, or may be later in coming. If after the time programmed in this register is met and graphics is still idle, then RC1(e)/RC6x is allowed to be entered.

0 = 0 usec
1 = 1.28 usec
2 = 2.56 usec
3 = 3.84 usec
FF FFFF = 21.474 sec

pmcr_rc_sleep[23:0]
# Unit Level Clock Gating Control 1

## UCGCTL1 - Unit Level Clock Gating Control 1

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31</td>
<td>Sarbunit Clock Gating Disable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SARB unit Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>0</td>
<td>30</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td>0</td>
<td>29</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td>0</td>
<td>28</td>
<td>ICunit Clock Gating Disable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ICunit Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>0</td>
<td>27</td>
<td>HIZunit Clock Gating Disable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HIZunit Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>0</td>
<td>26</td>
<td>GWunit Clock Gating Disable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GWunit Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>Register</td>
<td>Description</td>
<td>Default Value</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
<td>---------------</td>
</tr>
<tr>
<td>UCGCTL1</td>
<td>GTIunit Clock Gating Disable</td>
<td>1b</td>
</tr>
<tr>
<td></td>
<td>GSunit Clock Gating Disable</td>
<td></td>
</tr>
<tr>
<td></td>
<td>GPMunit Clock Gating Disable</td>
<td>1b</td>
</tr>
<tr>
<td></td>
<td>GAMunit Clock Gating Disable</td>
<td>0b</td>
</tr>
<tr>
<td></td>
<td>GACunit Clock Gating Disable</td>
<td>0b</td>
</tr>
</tbody>
</table>

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

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

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

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

**GACunit Clock Gating Disable**
- **Default Value:** 0b
- **Access:** R/W
- **Control:**
  - '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)
  - '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
## UCGCTL1 - Unit Level Clock Gating Control 1

<table>
<thead>
<tr>
<th>Register</th>
<th>Default Value</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
</table>
| **GABunit Clock Gating Disable** | 0b            | R/W      | '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
                  |               |          | '1': Clock Gating Disabled. (i.e., clocks are toggling, always)              |
| **FTunit Clock Gating Disable** |               | R/W      | '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
                  |               |          | '1': Clock Gating Disabled. (i.e., clocks are toggling, always)              |
| **FLunit Clock Gating Disable** |               | R/W      | '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
                  |               |          | '1': Clock Gating Disabled. (i.e., clocks are toggling, always)              |
| **EU_FPUunit Clock Gating Disable** |               | R/W      | '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
                  |               |          | '1': Clock Gating Disabled. (i.e., clocks are toggling, always)              |
| **EU_TCunit Clock Gating Disable** |               | R/W      | '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
                  |               |          | '1': Clock Gating Disabled. (i.e., clocks are toggling, always)              |
| **EU_EMunit Clock Gating Disable** |               | R/W      | '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
                  |               |          | '1': Clock Gating Disabled. (i.e., clocks are toggling, always)              |
### UCGCTL1 - Unit Level Clock Gating Control 1

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
<th>Access</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>EU_GAunit Clock Gating Disable</td>
<td>EU_GAunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td>R/W</td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>EUunit Clock Gating Disable</td>
<td>EUunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td>R/W</td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>SVLunit Clock Gating Disable</td>
<td>SVLunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td>R/W</td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>DTunit Clock Gating Disable</td>
<td>DTunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td>R/W</td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>DMunit Clock Gating Disable</td>
<td>DMunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td>R/W</td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>DGunit Clock Gating Disable</td>
<td>DGunit Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td>R/W</td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>
## UCGCTL1 - Unit Level Clock Gating Control 1

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
<th>Access</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 DAPunit Clock Gating Disable</td>
<td>DAPunit Clock Gating Disable Control: &lt;br&gt;‘0’ : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) &lt;br&gt;‘1’ : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td>R/W</td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) &lt;br&gt;‘1’ : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>7 CSunit Clock Gating Disable</td>
<td>CSunit Clock Gating Disable Control: &lt;br&gt;‘0’ : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) &lt;br&gt;‘1’ : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td>R/W</td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) &lt;br&gt;‘1’ : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>6 CLunit Clock Gating Disable</td>
<td>CLunit Clock Gating Disable Control: &lt;br&gt;‘0’ : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) &lt;br&gt;‘1’ : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td>R/W</td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) &lt;br&gt;‘1’ : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>5 BLBunit Clock Gating Disable</td>
<td>BLBunit Clock Gating Disable Control: &lt;br&gt;‘0’ : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) &lt;br&gt;‘1’ : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td>R/W</td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) &lt;br&gt;‘1’ : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>4 BFunit Clock Gating Disable</td>
<td>BFunit Clock Gating Disable Control: &lt;br&gt;‘0’ : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) &lt;br&gt;‘1’ : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td>R/W</td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) &lt;br&gt;‘1’ : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>3 BDunit Clock Gating Disable</td>
<td>BDunit Clock Gating Disable Control: &lt;br&gt;‘0’ : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) &lt;br&gt;‘1’ : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td>R/W</td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) &lt;br&gt;‘1’ : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>
## UCGCTL1 - Unit Level Clock Gating Control 1

<table>
<thead>
<tr>
<th></th>
<th>BCSunit Clock Gating Disable</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Access: R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td>BCSunit Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>AVSunit Clock Gating Disable</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Access: R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td>AVSunit Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>SPARE RAM Clock Gating Disable</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Access: R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SPARE RAM Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'0': RAM Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1': RAM Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
</tbody>
</table>
### Unit Level Clock Gating Control 1

#### UCGCTL1 - Unit Level Clock Gating Control 1

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31</td>
<td><strong>Sarbunit Clock Gating Disable</strong></td>
</tr>
<tr>
<td>30</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td>29</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td>28</td>
<td>27</td>
<td><strong>ICunit Clock Gating Disable</strong></td>
</tr>
<tr>
<td>27</td>
<td></td>
<td><strong>HIZunit Clock Gating Disable</strong></td>
</tr>
<tr>
<td>26</td>
<td></td>
<td><strong>GWunit Clock Gating Disable</strong></td>
</tr>
</tbody>
</table>

**Register Space:** MMIO: 0/2/0  
**Project:** CHV, BSW  
**Source:** PRM  
**Default Value:** 0x02F00000  
**Size (in bits):** 32  
**Address:** 09400h
<table>
<thead>
<tr>
<th>Register Name</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>UCGCTL1 - Unit Level Clock Gating Control 1</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>GTIunit Clock Gating Disable</strong></td>
<td>Default Value: 1b Access: R/W GTI Units Clock Gating Disable Control: '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>GSunit Clock Gating Disable</strong></td>
<td>Access: R/W GSunit Clock Gating Disable Control: '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>GPMunit Clock Gating Disable</strong></td>
<td>Default Value: 1b Access: R/W GPMunit Clock Gating Disable Control: '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>GAMunit Clock Gating Disable</strong></td>
<td>Default Value: 1b Access: R/W GAMunit Clock Gating Disable Control: '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>GACunit Clock Gating Disable</strong></td>
<td>Default Value: 1b Access: R/W GACunit Clock Gating Disable Control: '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>GABunit Clock Gating Disable</strong></td>
<td></td>
<td>Default Value: 1b</td>
<td></td>
</tr>
</tbody>
</table>
### UCGCTL1 - Unit Level Clock Gating Control 1

<table>
<thead>
<tr>
<th>Access:</th>
<th>R/W</th>
</tr>
</thead>
<tbody>
<tr>
<td>GABunit Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
</tbody>
</table>

**19** FTunit Clock Gating Disable

<table>
<thead>
<tr>
<th>Access:</th>
<th>R/W</th>
</tr>
</thead>
<tbody>
<tr>
<td>FTunit Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
</tbody>
</table>

**18** FLunit Clock Gating Disable

<table>
<thead>
<tr>
<th>Access:</th>
<th>R/W</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLunit Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
</tbody>
</table>

**17** EU_FPUunit Clock Gating Disable

<table>
<thead>
<tr>
<th>Access:</th>
<th>R/W</th>
</tr>
</thead>
<tbody>
<tr>
<td>EU_FPUunit Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
</tbody>
</table>

**16** EU_TCunit Clock Gating Disable

<table>
<thead>
<tr>
<th>Access:</th>
<th>R/W</th>
</tr>
</thead>
<tbody>
<tr>
<td>EU_TCunit Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
</tbody>
</table>

**15** EU_EMunit Clock Gating Disable

<table>
<thead>
<tr>
<th>Access:</th>
<th>R/W</th>
</tr>
</thead>
<tbody>
<tr>
<td>EU_EMunit Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
</tbody>
</table>
# UCGCTL1 - Unit Level Clock Gating Control 1

<table>
<thead>
<tr>
<th>Register Name</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EU_GAunit Clock Gating Disable</td>
<td>R/W</td>
<td>EU_GAunit Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>EUunit Clock Gating Disable</td>
<td>R/W</td>
<td>EUunit Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>SVLunit Clock Gating Disable</td>
<td>R/W</td>
<td>SVLunit Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>DTunit Clock Gating Disable</td>
<td>R/W</td>
<td>DTunit Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>DMunit Clock Gating Disable</td>
<td>R/W</td>
<td>DMunit Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>DGunit Clock Gating Disable</td>
<td>R/W</td>
<td>DGunit Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>
## UCGCTL1 - Unit Level Clock Gating Control 1

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
<th>Access</th>
<th>Control Detail</th>
</tr>
</thead>
</table>
| 8 DAPunit    | Clock Gating Disable                                  | R/W    | '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
              |            |                                   | '1' : Clock Gating Disabled. (i.e., clocks are toggling, always) |
| 7 CSunit     | Clock Gating Disable                                  | R/W    | '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
              |            |                                   | '1' : Clock Gating Disabled. (i.e., clocks are toggling, always) |
| 6 CLunit     | Clock Gating Disable                                  | R/W    | '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
              |            |                                   | '1' : Clock Gating Disabled. (i.e., clocks are toggling, always) |
| 5 BLBunit    | Clock Gating Disable                                  | R/W    | '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
              |            |                                   | '1' : Clock Gating Disabled. (i.e., clocks are toggling, always) |
| 4 BFunit     | Clock Gating Disable                                  | R/W    | '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
              |            |                                   | '1' : Clock Gating Disabled. (i.e., clocks are toggling, always) |
| 3 BDunit     | Clock Gating Disable                                  | R/W    | '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
              |            |                                   | '1' : Clock Gating Disabled. (i.e., clocks are toggling, always) |
## UCGCTL1 - Unit Level Clock Gating Control 1

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
<th>Control Description</th>
</tr>
</thead>
</table>
| 2   | BCSunit Clock Gating Disable            | R/W    | '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
|     |                                          |        | '1': Clock Gating Disabled. (i.e., clocks are toggling, always)                      |
| 1   | AVSunit Clock Gating Disable            | R/W    | '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
|     |                                          |        | '1': Clock Gating Disabled. (i.e., clocks are toggling, always)                      |
| 0   | SPARE RAM Clock Gating Disable          | R/W    | '0': RAM Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
|     |                                          |        | '1': RAM Clock Gating Disabled. (i.e., clocks are toggling, always)                  |
# Unit Level Clock Gating Control 2

<table>
<thead>
<tr>
<th>UCGCTRL2 - Unit Level Clock Gating Control 2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 09404h</td>
</tr>
</tbody>
</table>

Unit Level Clock Gating Control Registers. Refer to the Programming notes mentioned near GT Interface Registers in PRM

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31</td>
<td><strong>VFunit Clock Gating Disable</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>VFunit Clock Gating Disable Control:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>30</td>
<td></td>
<td><strong>VDSunit Clock Gating Disable</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>VDSunit Clock Gating Disable Control:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>29</td>
<td></td>
<td><strong>VDIunit Clock Gating Disable</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>VDIunit Clock Gating Disable Control:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>28</td>
<td></td>
<td><strong>VCSunit Clock Gating Disable</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>VCSunit Clock Gating Disable Control:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>
## UCGCTL2 - Unit Level Clock Gating Control 2

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
<th>Access</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>DTOunit Clock Gating Disable</td>
<td><a href="#">DTOunit Clock Gating Disable Control</a></td>
<td>R/W</td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td>VCPunit Clock Gating Disable</td>
<td><a href="#">VCPunit Clock Gating Disable Control</a></td>
<td>R/W</td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td>VCDunit Clock Gating Disable</td>
<td><a href="#">VCDunit Clock Gating Disable Control</a></td>
<td>R/W</td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td>URBMunit Clock Gating Disable</td>
<td><a href="#">URBMunit Clock Gating Disable Control</a></td>
<td>R/W</td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td>TSGunit Clock Gating Disable</td>
<td><a href="#">TSGunit Clock Gating Disable Control</a></td>
<td>R/W</td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td>TDLunit Clock Gating Disable</td>
<td><a href="#">TDLunit Clock Gating Disable Control</a></td>
<td>R/W</td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
</tbody>
</table>

**DTOunit Clock Gating Disable Control:**

- '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)
- '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)

**VCPunit Clock Gating Disable Control:**

- '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)
- '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)

**VCDunit Clock Gating Disable Control:**

- '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)
- '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)

**URBMunit Clock Gating Disable Control:**

- '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)
- '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)

**TSGunit Clock Gating Disable Control:**

- '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)
- '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)

**TDLunit Clock Gating Disable Control:**

- '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)
- '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)
## UCGCTL2 - Unit Level Clock Gating Control 2

<table>
<thead>
<tr>
<th>21</th>
<th>TDSunit Clock Gating Disable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>TDSunit Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>20</th>
<th>SVSMunit Clock Gating Disable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>SVSMunit Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>19</th>
<th>SVGunit Clock Gating Disable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>SVGunit Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>18</th>
<th>SOunit Clock Gating Disable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>SOunit Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>17</th>
<th>Slunit Clock Gating Disable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Slunit Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>16</th>
<th>SFunit Clock Gating Disable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>SFunit Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
</tbody>
</table>
### UCGCTRL2 - Unit Level Clock Gating Control 2

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
<th>Access</th>
<th>Control Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>15 SECunit Clock Gating Disable</td>
<td></td>
<td>R/W</td>
<td>'0': Clock Gating Enabled (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>14 SCunit Clock Gating Disable</td>
<td></td>
<td>R/W</td>
<td>'0': Clock Gating Enabled (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>13 RCZunit Clock Gating Disable</td>
<td></td>
<td>R/W</td>
<td>'0': Clock Gating Enabled (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>12 RCPBunit Clock Gating Disable</td>
<td></td>
<td>R/W</td>
<td>'0': Clock Gating Enabled (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>11 RCCunit Clock Gating Disable</td>
<td></td>
<td>R/W</td>
<td>'0': Clock Gating Enabled (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>10 QCunit Clock Gating Disable</td>
<td></td>
<td>R/W</td>
<td>'0': Clock Gating Enabled (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled (i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>
### UCGCTL2 - Unit Level Clock Gating Control 2

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
<th>Access</th>
<th>Control Details</th>
</tr>
</thead>
</table>
| **PSDunit Clock Gating Disable** | Access: R/W                                                                  |        | '0': Clock Gating Enabled (i.e., clocks can be gated when they are not required to toggle for functionality)  
|                           | '1': Clock Gating Disabled (i.e., clocks are toggling, always)               |        |                                                                                 |
| **PLunit Clock Gating Disable** | Access: R/W                                                                  |        | '0': Clock Gating Enabled (i.e., clocks can be gated when they are not required to toggle for functionality) 
|                           | '1': Clock Gating Disabled (i.e., clocks are toggling, always)               |        |                                                                                 |
| **MTunit Clock Gating Disable** | Access: R/W                                                                  |        | '0': Clock Gating Enabled (i.e., clocks can be gated when they are not required to toggle for functionality)  
|                           | '1': Clock Gating Disabled (i.e., clocks are toggling, always)               |        |                                                                                 |
| **MPCunit Clock Gating Disable** | Access: R/W                                                                  |        | '0': Clock Gating Enabled (i.e., clocks can be gated when they are not required to toggle for functionality) 
|                           | '1': Clock Gating Disabled (i.e., clocks are toggling, always)               |        |                                                                                 |
| **TDGunit Clock Gating Disable** | Access: R/W                                                                  |        | '0': Clock Gating Enabled (i.e., clocks can be gated when they are not required to toggle for functionality)  
|                           | '1': Clock Gating Disabled (i.e., clocks are toggling, always)               |        |                                                                                 |
| **MSCunit Clock Gating Disable** | Access: R/W                                                                  |        | '0': Clock Gating Enabled (i.e., clocks can be gated when they are not required to toggle for functionality)  
|                           | '1': Clock Gating Disabled (i.e., clocks are toggling, always)               |        |                                                                                 |
# UCGCTL2 - Unit Level Clock Gating Control 2

<table>
<thead>
<tr>
<th>Address</th>
<th>Description</th>
<th>Access</th>
<th>Control Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>TEunit Clock Gating Disable</td>
<td>R/W</td>
<td>'0': Clock Gating Enabled. ('i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. ('i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>2</td>
<td>TETGunit Clock Gating Disable</td>
<td>R/W</td>
<td>'0': Clock Gating Enabled. ('i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. ('i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>1</td>
<td>MAunit Clock Gating Disable</td>
<td>R/W</td>
<td>'0': Clock Gating Enabled. ('i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. ('i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>0</td>
<td>IZunit Clock Gating Disable</td>
<td>R/W</td>
<td>'0': Clock Gating Enabled. ('i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. ('i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>
## Unit Level Clock Gating Control 2

### UCGCTL2 - Unit Level Clock Gating Control 2

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td><strong>VFunit Clock Gating Disable</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>31</td>
<td><strong>VFunit Clock Gating Disable Control:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td></td>
<td><strong>VDSunit Clock Gating Disable</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>VDSunit Clock Gating Disable Control:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td></td>
<td><strong>VDIunit Clock Gating Disable</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>VDIunit Clock Gating Disable Control:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td></td>
<td><strong>VCSunit Clock Gating Disable</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>VCSunit Clock Gating Disable Control:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
</tbody>
</table>
### UCGCTL2 - Unit Level Clock Gating Control 2

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
</tr>
</thead>
</table>
| DTOunit Clock Gating Disable | DTOunit Clock Gating Disable Control:  
'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
'1' : Clock Gating Disabled. (i.e., clocks are toggling, always) |
| Access: R/W               |                                                                             |

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
</tr>
</thead>
</table>
| VCPunit Clock Gating Disable | VCPunit Clock Gating Disable Control:  
'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
'1' : Clock Gating Disabled. (i.e., clocks are toggling, always) |
| Access: R/W               |                                                                             |

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
</tr>
</thead>
</table>
| VCDunit Clock Gating Disable | VCDunit Clock Gating Disable Control:  
'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
'1' : Clock Gating Disabled. (i.e., clocks are toggling, always) |
| Access: R/W               |                                                                             |

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
</tr>
</thead>
</table>
| URBMunit Clock Gating Disable | URBMunit Clock Gating Disable Control:  
'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
'1' : Clock Gating Disabled. (i.e., clocks are toggling, always) |
| Access: R/W               |                                                                             |

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
</tr>
</thead>
</table>
| TSGunit Clock Gating Disable | TSGunit Clock Gating Disable Control:  
'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
'1' : Clock Gating Disabled. (i.e., clocks are toggling, always) |
| Access: R/W               |                                                                             |

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
</tr>
</thead>
</table>
| TDLunit Clock Gating Disable | TDLunit Clock Gating Disable Control:  
'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
'1' : Clock Gating Disabled. (i.e., clocks are toggling, always) |
| Access: R/W               |                                                                             |
## UCGCTL2 - Unit Level Clock Gating Control 2

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>Access</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>21</td>
<td>TDSunit Clock Gating Disable</td>
<td></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>TDSunit Clock Gating Disable Control:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>SVSMunit Clock Gating Disable</td>
<td></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>SVSMunit Clock Gating Disable Control:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>SVGunit Clock Gating Disable</td>
<td></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>SVGunit Clock Gating Disable Control:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>SOunit Clock Gating Disable</td>
<td></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>SOunit Clock Gating Disable Control:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Slunit Clock Gating Disable</td>
<td></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>Slunit Clock Gating Disable Control:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>SFunit Clock Gating Disable</td>
<td></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>SFunit Clock Gating Disable Control:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Unit Level Clock Gating Control 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----</td>
<td>-----------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td><strong>SECunit Clock Gating Disable</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
|     | SECunit Clock Gating Disable Control:  
|     | '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
|     | '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)  |
| 14  | **SCunit Clock Gating Disable**   |
|     | Access: R/W                       |
|     | SCunit Clock Gating Disable Control:  
|     | '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
|     | '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)  |
| 13  | **RCZunit Clock Gating Disable**  |
|     | Access: R/W                       |
|     | RCZunit Clock Gating Disable Control:  
|     | '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
|     | '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)  |
| 12  | **RCPBunit Clock Gating Disable** |
|     | Access: R/W                       |
|     | RCPBunit Clock Gating Disable Control:  
|     | '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
|     | '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)  |
| 11  | **RCCunit Clock Gating Disable**  |
|     | Access: R/W                       |
|     | RCCunit Clock Gating Disable Control:  
|     | '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
|     | '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)  |
| 10  | **QCunit Clock Gating Disable**   |
|     | Access: R/W                       |
|     | QCunit Clock Gating Disable Control:  
|     | '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
<p>|     | '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)  |</p>
<table>
<thead>
<tr>
<th></th>
<th>Register Name</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
</table>
| 9 | PSDunit Clock Gating Disable  | R/W    | **PSDunit Clock Gating Disable Control:**  
|    |                               |        | '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
|    |                               |        | '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)  |
| 8 | PLunit Clock Gating Disable   | R/W    | **PLunit Clock Gating Disable Control:**  
|    |                               |        | '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
|    |                               |        | '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)  |
| 7 | MTunit Clock Gating Disable   | R/W    | **MTunit Clock Gating Disable Control:**  
|    |                               |        | '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
|    |                               |        | '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)  |
| 6 | MPCunit Clock Gating Disable  | R/W    | **MPCunit Clock Gating Disable Control:**  
|    |                               |        | '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
|    |                               |        | '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)  |
| 5 | TDGunit Clock Gating Disable  | R/W    | **TDGunit Clock Gating Disable Control:**  
|    |                               |        | '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
|    |                               |        | '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)  |
| 4 | MSCunit Clock Gating Disable  | R/W    | **MSCunit Clock Gating Disable Control:**  
|    |                               |        | '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
|    |                               |        | '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)  |
## UCGCT2 - Unit Level Clock Gating Control 2

<table>
<thead>
<tr>
<th>Offset</th>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| 3      | TEunit Clock Gating Disable | Access: R/W  
TEunit Clock Gating Disable Control:  
'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
'1': Clock Gating Disabled. (i.e., clocks are toggling, always) |
| 2      | TETGunit Clock Gating Disable | Access: R/W  
TETGunit Clock Gating Disable Control:  
'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
'1': Clock Gating Disabled. (i.e., clocks are toggling, always) |
| 1      | MAunit Clock Gating Disable | Access: R/W  
MAunit Clock Gating Disable Control:  
'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
'1': Clock Gating Disabled. (i.e., clocks are toggling, always) |
| 0      | IZunit Clock Gating Disable | Access: R/W  
IZunit Clock Gating Disable Control:  
'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
'1': Clock Gating Disabled. (i.e., clocks are toggling, always) |
Unit Level Clock Gating Control 3

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31</td>
<td>Flunits 2nd Clock Gating Disable</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Flunits 2nd Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td></td>
<td>SVRRunits Clock Gating Disable</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SVRRunits' Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td></td>
<td>VCRunits Clock Gating Disable</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VCRunits' Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td></td>
<td>EDTunits Clock Gating Disable</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EDTunits' Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
</tbody>
</table>
### UCGCTRL3 - Unit Level Clock Gating Control 3

#### 27 VCUnit Clock Gating Disable

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

#### 26 HEVC DOP Gating Enable

**Default Value:** 1b  
**Project:** CHV, BSW  
**Access:** R/W  
**HEVCUNIT Clock Gating Enable Control:**
- '0': VIDPAR8/VIDPAR9 DOP gating is not affected by this bit (i.e., this bit does not affect functional DOP gating)
- '1': VIDPAR8/VIDPAR9 cmclk/cuclk DOP’s are gated. (i.e., functional clocks aren’t toggling, always)

#### 26 HEVC DOP Gating Enable

**Default Value:** 0b  
**Project:** CHV, BSW  
**Access:** R/W  
**HEVCUNIT Clock Gating Enable Control:**
- '0': VIDPAR8/VIDPAR9 DOP gating is not affected by this bit (i.e., this bit does not affect functional DOP gating)
- '1': VIDPAR8/VIDPAR9 cmclk/cuclk DOP’s are gated. (i.e., functional clocks aren’t toggling, always)

#### 25 HSunit Clock Gating Disable

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

#### 24 SOLunit Clock Gating Disable

**Access:** R/W  
**SOLunit Clock Gating Disable Control:**
- '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)
- '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
# UCGCTL3 - Unit Level Clock Gating Control 3

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
<th>Access</th>
<th>Control</th>
<th>Description</th>
</tr>
</thead>
</table>
| QRUnit Clock Gating Disable                   |                                                       | R/W    | QRCUnit Clock Gating Disable Control: | '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
'1' : Clock Gating Disabled. (i.e., clocks are toggling, always) |
| MSPBISTUnit Clock Gating Disable              |                                                       | R/W    | MSPBISTUnit Clock Gating Disable Control: | '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
'1' : Clock Gating Disabled. (i.e., clocks are toggling, always) |
| BSPUnit Clock Gating Disable                  |                                                       | R/W    | BSPUnit Clock Gating Disable Control: | '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
'1' : Clock Gating Disabled. (i.e., clocks are toggling, always) |
| SBEUnit Clock Gating Disable                  |                                                       | R/W    | SBEUnit Clock Gating Disable Control: | '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
'1' : Clock Gating Disabled. (i.e., clocks are toggling, always) |
| BCUnit Clock Gating Disable                   |                                                       | R/W    | BCUnit Clock Gating Disable Control: | '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
'1' : Clock Gating Disabled. (i.e., clocks are toggling, always) |
| WMBEUnit Clock Gating Disable                 |                                                       | R/W    | WMBEUnit Clock Gating Disable Control: | '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
'1' : Clock Gating Disabled. (i.e., clocks are toggling, always) |
# UCGCTRL3 - Unit Level Clock Gating Control 3

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
<th>Control</th>
</tr>
</thead>
</table>
| 16  | WMFEnit Clock Gating Disable          | R/W    | '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
|     |                                       |        | '1' : Clock Gating Disabled. (i.e., clocks are toggling, always) |
| 15  | VSCanit Clock Gating Disable          | R/W    | '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
|     |                                       |        | '1' : Clock Gating Disabled. (i.e., clocks are toggling, always) |
| 14  | Reserved                              |        |                              |
| 13  | USBunit Clock Gating Disable          | R/W    | '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
|     |                                       |        | '1' : Clock Gating Disabled. (i.e., clocks are toggling, always) |
| 12  | STCanit Clock Gating Disable          | R/W    | '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
|     |                                       |        | '1' : Clock Gating Disabled. (i.e., clocks are toggling, always) |
| 11  | VSunit Clock Gating Disable           | R/W    | '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
|     |                                       |        | '1' : Clock Gating Disabled. (i.e., clocks are toggling, always) |
| 10  | VOPunit Clock Gating Disable          | R/W    | '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
|     |                                       |        | '1' : Clock Gating Disabled. (i.e., clocks are toggling, always) |
### UCGCTL3 - Unit Level Clock Gating Control 3

<table>
<thead>
<tr>
<th></th>
<th>VMXunit Clock Gating Disable</th>
<th></th>
<th>VMXunit Clock Gating Disable Control:</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>Access: R/W</td>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>VMEunit Clock Gating Disable</th>
<th></th>
<th>VMEunit Clock Gating Disable Control:</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>Access: R/W</td>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>VMDunit Clock Gating Disable</th>
<th></th>
<th>VMDunit Clock Gating Disable Control:</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>Access: R/W</td>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>VMCunit Clock Gating Disable</th>
<th></th>
<th>VMCunit Clock Gating Disable Control:</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>Access: R/W</td>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>VLFunit Clock Gating Disable</th>
<th></th>
<th>VLFunit Clock Gating Disable Control:</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Access: R/W</td>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>VITunit Clock Gating Disable</th>
<th></th>
<th>VITunit Clock Gating Disable Control:</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Access: R/W</td>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>
### UCGCTL3 - Unit Level Clock Gating Control 3

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td><strong>VIPunit Clock Gating Disable</strong></td>
<td>R/W</td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>2</td>
<td><strong>VINunit Clock Gating Disable</strong></td>
<td>R/W</td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>1</td>
<td><strong>VFTunit Clock Gating Disable</strong></td>
<td>R/W</td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>0</td>
<td><strong>VFEunit Clock Gating Disable</strong></td>
<td>R/W</td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>
# Unit Level Clock Gating Control 3

## UCGCTL3 - Unit Level Clock Gating Control 3

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x04000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>09408h</td>
</tr>
</tbody>
</table>

Unit Level Clock Gating Control Registers.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Access:</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31</td>
<td><strong>Flunits 2nd Clock Gating Disable</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Flunits 2nd Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td></td>
<td><strong>SVRRunit Clock Gating Disable</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SVRRunits' Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td></td>
<td><strong>VCRunit Clock Gating Disable</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VCRunits' Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td></td>
<td><strong>EDTunit Clock Gating Disable</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EDTunits' Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
</tbody>
</table>
## UCGCTL3 - Unit Level Clock Gating Control 3

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
<th>Access</th>
<th>Default Value</th>
<th>Notes</th>
</tr>
</thead>
</table>
| **27 VClunit Clock Gating Disable** | VClunits' Clock Gating Disable Control:  
  '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
  '1' : Clock Gating Disabled. (i.e., clocks are toggling, always) | R/W    |               |                                                                      |
| **26 2x Assign fub XOR Clock Gating Disable** | 2x Assign fub XOR Clock Gating Disable Control:  
  '0' : 2x Assign fub XOR Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
  '1' : 2x Assign fub XOR Clock Gating Disabled. (i.e., clocks are toggling, always) | R/W    | 1b            |                                                                      |
| **25 HSunit Clock Gating Disable** | HSunit Clock Gating Disable Control:  
  '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
  '1' : Clock Gating Disabled. (i.e., clocks are toggling, always) | R/W    |               |                                                                      |
| **24 SOLunit Clock Gating Disable** | SOLunit Clock Gating Disable Control:  
  '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
  '1' : Clock Gating Disabled. (i.e., clocks are toggling, always) | R/W    |               |                                                                      |
| **23 QRCunit Clock Gating Disable** | QRCunit Clock Gating Disable Control:  
  '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
  '1' : Clock Gating Disabled. (i.e., clocks are toggling, always) | R/W    |               |                                                                      |
| **22 MSPBISTunit Clock Gating Disable** | MSPBISTunit Clock Gating Disable Control:  
  '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
  '1' : Clock Gating Disabled. (i.e., clocks are toggling, always) | R/W    |               |                                                                      |
### UCGCTL3 - Unit Level Clock Gating Control 3

<table>
<thead>
<tr>
<th>Register</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>21 BSPunit Clock Gating Disable</strong></td>
<td>R/W</td>
<td>Access: R/W&lt;br&gt;<strong>BSPunit Clock Gating Disable Control:</strong>&lt;br&gt;‘0’: Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)&lt;br&gt;‘1’: Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td><strong>20 OACSunit Clock Gating Disable</strong></td>
<td>R/W</td>
<td>Access: R/W&lt;br&gt;<strong>OACSunit Clock Gating Disable Control:</strong>&lt;br&gt;‘0’: Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)&lt;br&gt;‘1’: Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td><strong>19 SBEunit Clock Gating Disable</strong></td>
<td>R/W</td>
<td>Access: R/W&lt;br&gt;<strong>SBEunit Clock Gating Disable Control:</strong>&lt;br&gt;‘0’: Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)&lt;br&gt;‘1’: Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td><strong>18 BCunit Clock Gating Disable</strong></td>
<td>R/W</td>
<td>Access: R/W&lt;br&gt;<strong>BCunit Clock Gating Disable Control:</strong>&lt;br&gt;‘0’: Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)&lt;br&gt;‘1’: Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td><strong>17 WMBE Clock Gating Disable</strong></td>
<td>R/W</td>
<td>Access: R/W&lt;br&gt;<strong>WMBEunit Clock Gating Disable Control:</strong>&lt;br&gt;‘0’: Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)&lt;br&gt;‘1’: Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td><strong>16 WMFEunit Clock Gating Disable</strong></td>
<td>R/W</td>
<td>Access: R/W&lt;br&gt;<strong>WMFEunit Clock Gating Disable Control:</strong>&lt;br&gt;‘0’: Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)&lt;br&gt;‘1’: Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>Register Name</td>
<td>Access</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>UCGCTL3 - Unit Level Clock Gating Control 3</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15 VSCunit Clock Gating Disable</td>
<td>R/W</td>
<td>VSCunit Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>‘0’ : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>‘1’ : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>14 Reserved</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>13 USBunit Clock Gating Disable</td>
<td>R/W</td>
<td>USBunit Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>‘0’ : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>‘1’ : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>12 STCunit Clock Gating Disable</td>
<td>R/W</td>
<td>STCunit Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>‘0’ : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>‘1’ : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>11 VSunit Clock Gating Disable</td>
<td>R/W</td>
<td>VSunit Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>‘0’ : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>‘1’ : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>10 VOPunit Clock Gating Disable</td>
<td>R/W</td>
<td>VOPunit Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>‘0’ : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>‘1’ : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>
## UCGCTL3 - Unit Level Clock Gating Control 3

<table>
<thead>
<tr>
<th></th>
<th>VMXunit Clock Gating Disable</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td><strong>VMXunit Clock Gating Disable</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>VMXunit Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>VMEunit Clock Gating Disable</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td><strong>VMEunit Clock Gating Disable</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>VMEunit Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>VMDunit Clock Gating Disable</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td><strong>VMDunit Clock Gating Disable</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>VMDunit Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>VMCunit Clock Gating Disable</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td><strong>VMCunit Clock Gating Disable</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>VMCunit Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>VLFunit Clock Gating Disable</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td><strong>VLFunit Clock Gating Disable</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>VLFunit Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>VITunit Clock Gating Disable</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td><strong>VITunit Clock Gating Disable</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>VITunit Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
</tbody>
</table>
## UCGCTL3 - Unit Level Clock Gating Control 3

<table>
<thead>
<tr>
<th>Address</th>
<th>Description</th>
<th>Access</th>
<th>Control Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>VIPunit Clock Gating Disable</td>
<td>R/W</td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>2</td>
<td>VINunit Clock Gating Disable</td>
<td>R/W</td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>1</td>
<td>VFTunit Clock Gating Disable</td>
<td>R/W</td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>0</td>
<td>VFEunit Clock Gating Disable</td>
<td>R/W</td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>
## UCGCTL4 - Unit Level Clock Gating Control 4

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:30</td>
<td><strong>Reserved</strong>&lt;br&gt;Access: RO&lt;br&gt;Reserved.</td>
</tr>
<tr>
<td>29</td>
<td></td>
<td><strong>GAFSRRB unit Clock Gate Disable</strong>&lt;br&gt;Access: R/W&lt;br&gt;GAFSRRB units Clock Gating Disable Control:&lt;br&gt;'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)&lt;br&gt;'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>28</td>
<td></td>
<td><strong>RAMDFT units Clock Gate Disable</strong>&lt;br&gt;Access: R/W&lt;br&gt;RAMDFT units Clock Gating Disable Control:&lt;br&gt;'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)&lt;br&gt;'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>27</td>
<td></td>
<td><strong>TDCunit Clock Gate Disable</strong>&lt;br&gt;Access: R/W&lt;br&gt;TDCunit Clock Gating Disable Control:&lt;br&gt;'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)&lt;br&gt;'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>26</td>
<td></td>
<td><strong>L3 CBR 1x Clock Gate Disable</strong>&lt;br&gt;Access: R/W&lt;br&gt;L3 CBR units 1x Clock Gating Disable Control:&lt;br&gt;'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)&lt;br&gt;'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>
### UCGCTRL4 - Unit Level Clock Gating Control 4

<table>
<thead>
<tr>
<th>25</th>
<th>L3 BANK 2x Clock Gate Disable</th>
<th>Access: R/W</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>L3 BANK units 2x Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>24</th>
<th>L3 BANK 1x Clock Gate Disable</th>
<th>Access: R/W</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>L3 BANK units 1x Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>23</th>
<th>MBGFunit Clock Gate Disable</th>
<th>Default Value: 1b</th>
<th>Access: R/W</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MBGFunits Clock Gating Disable Control:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>22</th>
<th>MSQDunit 2x Clock Gate Disable</th>
<th>Default Value: 1b</th>
<th>Access: R/W</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MSQD units cu2x Clock Gating Disable Control:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>21</th>
<th>MSQDunit Clock Gate Disable</th>
<th>Default Value: 1b</th>
<th>Access: R/W</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MSQD units 1x Clock Gating Disable Control:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>20</th>
<th>MISDunits 2x Clock Gate Disable</th>
<th>Default Value: 1b</th>
<th>Access: R/W</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MISDunits cu2x Clock Gating Disable Control:</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## UCGCTL4 - Unit Level Clock Gating Control 4

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
<th>Default</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>UCGCTL4</td>
<td>Unit Level Clock Gating Control 4</td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>MISDunit Clock Gate Disable</td>
<td></td>
<td>R/W</td>
</tr>
<tr>
<td>Default Value:</td>
<td>1b</td>
<td>Access:</td>
<td></td>
</tr>
<tr>
<td>GAFMunits 1x Clock Gating Disable Control:</td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>GAFMunit Clock Gate Disable</td>
<td></td>
<td>R/W</td>
</tr>
<tr>
<td>Access:</td>
<td></td>
<td>GAFMunits Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>GAPCunit Clock Gate Disable</td>
<td></td>
<td>R/W</td>
</tr>
<tr>
<td>Access:</td>
<td></td>
<td>GAPCunits Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>GAPZunit Clock Gate Disable</td>
<td></td>
<td>R/W</td>
</tr>
<tr>
<td>Access:</td>
<td></td>
<td>GAPZunits Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>GAPL3unit Clock Gate Disable</td>
<td></td>
<td>R/W</td>
</tr>
<tr>
<td>Access:</td>
<td></td>
<td>GAPL3 units Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Register Name</td>
<td>Access</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>-------------------------------</td>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td><strong>UCGCTL4 - Unit Level Clock Gating Control 4</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>14 GAFSunit Clock Gate Disable</strong></td>
<td>R/W</td>
<td>GAFSunits’ Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
<tr>
<td><strong>13 GAHSunit Clock Gate Disable</strong></td>
<td>R/W</td>
<td>GAHSunits’ Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
<tr>
<td><strong>12 VIStunit Clock Gate Disable</strong></td>
<td>R/W</td>
<td>VIStunits’ Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
<tr>
<td><strong>11 VACunit Clock Gate Disable</strong></td>
<td>R/W</td>
<td>VACunits’ Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
<tr>
<td><strong>10 VAMunit Clock Gate Disable</strong></td>
<td>R/W</td>
<td>VAMunits Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
<tr>
<td><strong>9 VADunit Clock Gate Disable</strong></td>
<td>R/W</td>
<td>VADunits Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
</tbody>
</table>
### UCGCTL4 - Unit Level Clock Gating Control 4

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
<th>Access</th>
<th>Control Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>JPGunit Clock Gating Disable</td>
<td>JPGunits Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>VBPunits Clock Gating Disable</td>
<td>VBPunits Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>VHRunit Clock Gating Disable</td>
<td>VHRunits Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>VID4 VINunit Clock Gating Disable</td>
<td>VID4 VINunits' Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>VID3 VINunit Clock Gating Disable</td>
<td>VID3 VINunits’ Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>VID2 VINunit Clock Gating Disable</td>
<td>VID2 VINunits’ Clock Gating Disable Control: '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td>R/W</td>
<td></td>
</tr>
</tbody>
</table>
## UCGCTL4 - Unit Level Clock Gating Control 4

<table>
<thead>
<tr>
<th>2</th>
<th><strong>VID1 VINunits Clock Gating Disable</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td>0</td>
<td><strong>VID1 Vinunits’ Clock Gating Disable Control:</strong></td>
</tr>
<tr>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1:0</th>
<th><strong>MSQCunit Clock Gating Disable</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Default Value:</strong> 11b</td>
</tr>
<tr>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td><strong>MSQCunits’ Clock Gating Disable Control:</strong></td>
</tr>
<tr>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>
# Unit Level Clock Gating Control 4

## UCGCTL4 - Unit Level Clock Gating Control 4

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00F80003</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>0940Ch</td>
</tr>
</tbody>
</table>

Unit Level Clock Gating Control Registers.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:30</td>
<td><strong>Reserved</strong></td>
<td><strong>RO</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>rsvd</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td></td>
<td><strong>GAFSRRB unit Clock Gate Disable</strong></td>
<td><strong>R/W</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>GAFSRRB units Clock Gating Disable Control:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td></td>
<td><strong>RAMDFT units Clock Gate Disable</strong></td>
<td><strong>R/W</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>RAMDFT units Clock Gating Disable Control:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td></td>
<td><strong>L3 CBR 2x Clock Gate Disable</strong></td>
<td><strong>R/W</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>L3 CBR units 2x Clock Gating Disable Control:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td></td>
<td><strong>L3 CBR 1x Clock Gate Disable</strong></td>
<td><strong>R/W</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>L3 CBR units 1x Clock Gating Disable Control:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## UCGCTL4 - Unit Level Clock Gating Control 4

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>L3 BANK 2x Clock Gate Disable</td>
<td>L3 BANK units 2x Clock Gating Disable Control: '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>L3 BANK 1x Clock Gate Disable</td>
<td>L3 BANK units 1x Clock Gating Disable Control: '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>MBGFunc unit Clock Gate Disable</td>
<td>MBGFunc unit Clock Gating Disable Control: '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td>1b</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>MSQDunit 2x Clock Gate Disable</td>
<td>MSQD units cu2x Clock Gating Disable Control: '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td>1b</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>MSQDunit Clock Gate Disable</td>
<td>MSQD units 1x Clock Gating Disable Control: '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>MISDunits 2x Clock Gate Disable</td>
<td>MISDunits cu2x Clock Gating Disable Control: '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td>1b</td>
<td>R/W</td>
<td></td>
</tr>
</tbody>
</table>
## UCGCTL4 - Unit Level Clock Gating Control 4

<table>
<thead>
<tr>
<th>19</th>
<th>MISDunit Clock Gate Disable</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Default Value: 1b</td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td>MISDunits 1x Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>18</th>
<th>GAFMunit Clock Gate Disable</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td>GAFMunits Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>17</th>
<th>GAPCunit Clock Gate Disable</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td>GAPCunits Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>16</th>
<th>GAPZunit Clock Gate Disable</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td>GAPZunits Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>15</th>
<th>GAPL3unit Clock Gate Disable</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td>GAPL3 units Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>14</th>
<th>GAFSunit Clock Gate Disable</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td>GAFSunits Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>
### UCGCTL4 - Unit Level Clock Gating Control 4

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
</tr>
</thead>
</table>
| **13** GAHSunit Clock Gate Disable | **Access:** R/W  
GAHSunits’ Clock Gating Disable Control:  
'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
'1': Clock Gating Disabled. (i.e., clocks are toggling, always) |
| **12** VISunit Clock Gate Disable | **Access:** R/W  
VISunits’ Clock Gating Disable Control:  
'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
'1': Clock Gating Disabled. (i.e., clocks are toggling, always) |
| **11** VACunit Clock Gate Disable | **Access:** R/W  
VACunits’ Clock Gating Disable Control:  
'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
'1': Clock Gating Disabled. (i.e., clocks are toggling, always) |
| **10** VAMunit Clock Gate Disable | **Access:** R/W  
VAMunits Clock Gating Disable Control:  
'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
'1': Clock Gating Disabled. (i.e., clocks are toggling, always) |
| **9** VADuit Clock Gating Disable | **Access:** R/W  
VADunits Clock Gating Disable Control:  
'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
'1': Clock Gating Disabled. (i.e., clocks are toggling, always) |
| **8** JPGunit Clock Gating Disable | **Access:** R/W  
JPGunits Clock Gating Disable Control:  
'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
'1': Clock Gating Disabled. (i.e., clocks are toggling, always) |
### UCGCTL4 - Unit Level Clock Gating Control 4

<table>
<thead>
<tr>
<th></th>
<th>VBPunits Clock Gating Disable</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td>VBPunits Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>VHRunit Clock Gating Disable</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td>VHRunits Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>VID4 VINunit Clock Gating Disable</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td>VID4 VINunits’ Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>VID3 VINunit Clock Gating Disable</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td>VID3 VINunits’ Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>VID2 VINunit Clock Gating Disable</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td>VID2 VINunits’ Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>VID1 VINunit Clock Gating Disable</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td>VID1 VINunits’ Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>Register</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>UCGCTL4</td>
<td>Unit Level Clock Gating Control 4</td>
</tr>
<tr>
<td><strong>1:0</strong></td>
<td><strong>MSQCunit Clock Gating Disable</strong></td>
</tr>
<tr>
<td>Default Value:</td>
<td>11b</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
</tbody>
</table>

MSQCunits' Clock Gating Disable Control:
- '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)
- '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
### UCGCTL5 - Unit Level Clock Gating Control 5

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31</td>
<td>VCOPunit clock gating disable bit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WVCOP units Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>30</td>
<td></td>
<td>VMBunit clock gate disable bit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VMB units Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>29</td>
<td></td>
<td>VDMunit clock gate disable bit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VDM units Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>28</td>
<td></td>
<td>L3BANK unit culkgating disable bit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>L3bank units culk Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>27</td>
<td></td>
<td>L3BANK cu2x clock gate disable bit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>L3BANK units cu2x Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>
**UCGCTL5 - Unit Level Clock Gating Control 5**

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>LNunit clock gate disable bit</td>
<td>LNI units Clock Gating Disable Control: '0' : Clock Gating Enabled. '1' : Clock Gating Disabled.</td>
<td>R/W</td>
</tr>
<tr>
<td>LNEUNIT clock gate disable bit</td>
<td>LNE units Clock Gating Disable Control: '0' : Clock Gating Enabled. '1' : Clock Gating Disabled.</td>
<td>R/W</td>
</tr>
<tr>
<td>VVPunit clock gate disable bit</td>
<td>VVP units Clock Gating Disable Control: '0' : Clock Gating Enabled. '1' : Clock Gating Disabled.</td>
<td>R/W</td>
</tr>
<tr>
<td>WVFT unit clock gate disable bits</td>
<td>WVFT units Clock Gating Disable Control: '0' : Clock Gating Enabled. '1' : Clock Gating Disabled.</td>
<td>R/W</td>
</tr>
<tr>
<td>WBPS unit clock gate disable bit</td>
<td>WBPS units Clock Gating Disable Control: '0' : Clock Gating Enabled. '1' : Clock Gating Disabled.</td>
<td>R/W</td>
</tr>
<tr>
<td>WVMX unit clock gate disable bit</td>
<td>WVMX units Clock Gating Disable Control: '0' : Clock Gating Enabled. '1' : Clock Gating Disabled.</td>
<td>R/W</td>
</tr>
</tbody>
</table>
## UCGCTRL5 - Unit Level Clock Gating Control 5

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
<th>Access</th>
<th>Control Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 WVIP unit clock gate disable bit</td>
<td>WVIP unit Clock Gating Disable Control: '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>19 WVIT unit clock gate disable bit</td>
<td>WVIT units Clock Gating Disable Control: '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>18 WVIS unit clock gate disable</td>
<td>WVIS units Clock Gating Disable Control: '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>17 RPM units clock gate disable</td>
<td>RPM units Clock Gating Disable Control: '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>16 Reserved</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15 VECS unit clock gate disable</td>
<td>VECS units Clock Gating Disable Control: '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>14 GAHSV unit clock gate disable</td>
<td>GAHSV units Clock Gating Disable Control: '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>Register</td>
<td>Description</td>
<td>Access</td>
<td>Control Details</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
<td>--------</td>
<td>-----------------</td>
</tr>
<tr>
<td>13</td>
<td>GAHSD unit clock gate disable</td>
<td>R/W</td>
<td>'0': Clock Gating Enabled. ('i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. ('i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>12</td>
<td>GAV unit's clock gate disable</td>
<td>R/W</td>
<td>'0': Clock Gating Enabled. ('i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. ('i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>11</td>
<td>RSunit's clock gate disable</td>
<td>R/W</td>
<td>'0': Clock Gating Enabled. ('i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. ('i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>10</td>
<td>VFW units clock gate disable</td>
<td>R/W</td>
<td>'0': Clock Gating Enabled. ('i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. ('i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>9</td>
<td>VCW unit's clock gate disable</td>
<td>R/W</td>
<td>'0': Clock Gating Enabled. ('i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. ('i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>8</td>
<td>VEO unit's clock gate disable</td>
<td>R/W</td>
<td>'0': Clock Gating Enabled. ('i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. ('i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>
## UCGCTL5 - Unit Level Clock Gating Control 5

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
<th>Access</th>
<th>Control Details</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>VDN unit's clock gate disable</strong></td>
<td>Access: R/W</td>
<td>VDN units Clock Gating Disable Control: '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
<tr>
<td><strong>VTQunit's clock gate disable</strong></td>
<td>Access: R/W</td>
<td>VTQunits Clock Gating Disable Control: '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
<tr>
<td><strong>VPRunit's clock gate disable</strong></td>
<td>Access: R/W</td>
<td>VPR units Clock Gating Disable Control: '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
<tr>
<td><strong>IMEunit's clock gate disable</strong></td>
<td>Access: R/W</td>
<td>IME units Clock Gating Disable Control: '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
<tr>
<td><strong>CREunit clock gate disable</strong></td>
<td>Access: R/W</td>
<td>CRE units Clock Gating Disable Control: '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
<tr>
<td><strong>GAPSL unit clock gate disable</strong></td>
<td>Access: R/W</td>
<td>GAPSL units Clock Gating Disable Control: '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
</tbody>
</table>
### UCGCTL5 - Unit Level Clock Gating Control 5

<table>
<thead>
<tr>
<th></th>
<th>GAPSU Clock gate disable</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Access: R/W</td>
<td></td>
</tr>
</tbody>
</table>

GAPSU units Clock Gating Disable Control:
- '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)
- '1': Clock Gating Disabled. (i.e., clocks are toggling, always)

<table>
<thead>
<tr>
<th></th>
<th>SPMunit Clock gate disable</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Access: R/W</td>
<td></td>
</tr>
</tbody>
</table>

SPM units Clock Gating Disable Control:
- '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)
- '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
## Unit Level Clock Gating Control 5

**UCGCTL5 - Unit Level Clock Gating Control 5**

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31</td>
<td>VCOP unit clock gating disable bit</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VCOP units Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>

| 30    |     | VMB unit clock gating disable bit |
|       |     | **Access:** R/W |
|       |     | VMB units Clock Gating Disable Control: |
|       |     | '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) |
|       |     | '1' : Clock Gating Disabled. (i.e., clocks are toggling, always) |

| 29    |     | VDM unit clock gating disable bit |
|       |     | **Access:** R/W |
|       |     | VDM units Clock Gating Disable Control: |
|       |     | '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) |
|       |     | '1' : Clock Gating Disabled. (i.e., clocks are toggling, always) |

| 28    |     | L3BANK unit clock gating disable bit |
|       |     | **Access:** R/W |
|       |     | L3BANK units clock gating Disable Control: |
|       |     | '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) |
|       |     | '1' : Clock Gating Disabled. (i.e., clocks are toggling, always) |

<p>| 27    |     | L3BANK cu2x clock gating disable bit |
|       |     | <strong>Access:</strong> R/W |
|       |     | L3BANK units cu2x Clock Gating Disable Control: |
|       |     | '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) |
|       |     | '1' : Clock Gating Disabled. (i.e., clocks are toggling, always) |</p>
<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
<th>Access</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>UCGCTL5</td>
<td>Unit Level Clock Gating Control 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>26 LNIunit clock gate disable bit</td>
<td>Access: R/W</td>
<td></td>
<td>LNI units Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>25 LNEUNIT clock gate disable bit</td>
<td>Access: R/W</td>
<td></td>
<td>LNE units Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>24 VVPunit clock gate disable bit</td>
<td>Access: R/W</td>
<td></td>
<td>VVP units Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>23 WVFT unit clock gate disable bits</td>
<td>Access: R/W</td>
<td></td>
<td>WVFT units Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>22 WBPS unit clock gate disable bit</td>
<td>Access: R/W</td>
<td></td>
<td>WBPS units Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>21 WVMX unit clock gate disable bit</td>
<td>Access: R/W</td>
<td></td>
<td>WVMX units Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>
## UCGCTRL5 - Unit Level Clock Gating Control 5

<table>
<thead>
<tr>
<th></th>
<th>WVIP unit clock gate disable bit</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>20</strong></td>
<td>WVIP unit clock gate disable bit</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>WVIP unit Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>WVIT unit clock gate disable bit</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>19</strong></td>
<td>WVIT unit clock gate disable bit</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>WVIT units Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>WVIS unit clock gate disable</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>18</strong></td>
<td>WVIS unit clock gate disable</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>WVIS units Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>RPM units clock gate disable</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>17</strong></td>
<td>RPM units clock gate disable</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>RPM units Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>OASC unit clock gating disable</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>16</strong></td>
<td>OASC unit clock gating disable</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>OASC units Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>VECS unit clock gate disable</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>15</strong></td>
<td>VECS unit clock gate disable</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>VECS units Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
<tr>
<td>Register</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>UCGCTRL5</td>
<td>Unit Level Clock Gating Control 5</td>
</tr>
<tr>
<td>14 GAHSV</td>
<td>Unit clock gate disable</td>
</tr>
<tr>
<td>13 GAHSD</td>
<td>Unit clock gate disable</td>
</tr>
<tr>
<td>12 GAV</td>
<td>Unit's clock gate disable</td>
</tr>
<tr>
<td>11 RSW</td>
<td>Unit's clock gate disable</td>
</tr>
<tr>
<td>10 VFW</td>
<td>Units clock gate disable</td>
</tr>
<tr>
<td>9 VCW</td>
<td>Unit's clock gate disable</td>
</tr>
</tbody>
</table>
### UCGCTL5 - Unit Level Clock Gating Control 5

<table>
<thead>
<tr>
<th>8</th>
<th>VEO unit's clock gate disable</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Access:</strong> R/W</td>
<td></td>
</tr>
<tr>
<td>VEO units Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>7</th>
<th>VDN unit's clock gate disable</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Access:</strong> R/W</td>
<td></td>
</tr>
<tr>
<td>VDN units Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>6</th>
<th>VTQ unit's clock gate disable</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Access:</strong> R/W</td>
<td></td>
</tr>
<tr>
<td>VTQ units Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>5</th>
<th>VPR unit's clock gate disable</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Access:</strong> R/W</td>
<td></td>
</tr>
<tr>
<td>VPR units Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>4</th>
<th>IME unit's clock gate disable</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Access:</strong> R/W</td>
<td></td>
</tr>
<tr>
<td>IME units Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>3</th>
<th>CRE unit clock gate disable</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Access:</strong> R/W</td>
<td></td>
</tr>
<tr>
<td>CRE units Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
</tbody>
</table>
**UCGCTL5 - Unit Level Clock Gating Control 5**

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>GAPSL unit clock gate disable</td>
<td>R/W</td>
</tr>
<tr>
<td>1</td>
<td>GAPSU Clock gate disable</td>
<td>R/W</td>
</tr>
<tr>
<td>0</td>
<td>SPMunit Clock gate disable</td>
<td>R/W</td>
</tr>
</tbody>
</table>

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

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

**SPMunit Clock gate disable**
- Access: R/W
- SPM units Clock Gating Disable Control:
  - '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)
  - '1': Clock Gating Disabled. (i.e., clocks are toggling, always)
# Unit Level Clock Gating Control 6

## UCGCTL6 - Unit Level Clock Gating Control 6

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>09430h</td>
</tr>
</tbody>
</table>

### Unit Level Clock Gating Disable bits

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>30:28</td>
<td><strong>HDCunit clock gate disable</strong>&lt;br&gt;Access: R/W&lt;br&gt;HDC units Clock Gating Disable Control:&lt;br&gt;'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)&lt;br&gt;'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>27</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td>26</td>
<td></td>
<td><strong>GACVunit cuclk gate disable</strong>&lt;br&gt;Access: R/W&lt;br&gt;GACV units Clock Gating Disable Control:&lt;br&gt;'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)&lt;br&gt;'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>25</td>
<td></td>
<td><strong>GACBunit clock gate disable</strong>&lt;br&gt;Access: R/W&lt;br&gt;GACB units Clock Gating Disable Control:&lt;br&gt;'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)&lt;br&gt;'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>24</td>
<td></td>
<td><strong>GAPSunit clock gate disable</strong>&lt;br&gt;Access: R/W&lt;br&gt;GAPS units Clock Gating Disable Control:&lt;br&gt;'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)&lt;br&gt;'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>
### UCGCTL6 - Unit Level Clock Gating Control 6

<table>
<thead>
<tr>
<th>Bit</th>
<th>Field Description</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
</table>
| 23  | GGMTunit clock gate disable | R/W | '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
  '1' : Clock Gating Disabled. (i.e., clocks are toggling, always) |
| 22  | Reserved           |       |             |
| 21  | Reserved           |       |             |
| 20  | Reserved           |       |             |
| 19  | GACVunit clock gate disable | R/W | '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
  '1' : Clock Gating Disabled. (i.e., clocks are toggling, always) |
| 18  | BDMunit clock gate disable | R/W | '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
  '1' : Clock Gating Disabled. (i.e., clocks are toggling, always) |
| 17  | GATSunit clock gate disable | R/W | '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
  '1' : Clock Gating Disabled. (i.e., clocks are toggling, always) |
| 16  | Reserved           |       |             |
| 15  | STunit clock gate disable | R/W | '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
  '1' : Clock Gating Disabled. (i.e., clocks are toggling, always) |
<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
<th>Access</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDEunit clock gate disable</td>
<td>DE units Clock Gating Disable Control: '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td>R/W</td>
<td>Due to SDEUNIT bug HSD#1802092, this bit should be programmed to a 1 for CHV, BSW</td>
</tr>
<tr>
<td>VIN(VID6) unit clock gate disable</td>
<td>VIN(VID6) units Clock Gating Disable Control: '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>VIN(VID5) unit clock gate disable</td>
<td>VIN(VID5) units Clock Gating Disable Control: '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>WVOPunit clock gate disable</td>
<td>WVOP units Clock Gating Disable Control: '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>WUSB unit clock gate disable</td>
<td>WUSB units Clock Gating Disable Control: '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td>WSECunit clock gate disable</td>
<td>WSEC units Clock Gating Disable Control: '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td>R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Description</td>
<td>Access</td>
<td>Description</td>
</tr>
<tr>
<td>---</td>
<td>----------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>8</td>
<td><strong>WRS unit clock gate disable</strong></td>
<td>R/W</td>
<td>WRS units Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>7</td>
<td><strong>WQRC unit clock gate disable</strong></td>
<td>R/W</td>
<td>WQRC units Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>6</td>
<td><strong>WMPC unit level clock gate disable</strong></td>
<td>R/W</td>
<td>WMPC units Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>5</td>
<td><strong>WIN unit clock gate disable</strong></td>
<td>R/W</td>
<td>WIN units Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>4</td>
<td><strong>WIME unit clock gate disable</strong></td>
<td>R/W</td>
<td>WIME units Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>3</td>
<td><strong>WHME unit clock gate disable</strong></td>
<td>R/W</td>
<td>WHME units Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>Offset</td>
<td>Description</td>
<td>Access</td>
<td>Control Details</td>
</tr>
<tr>
<td>--------</td>
<td>--------------------------------------------------</td>
<td>--------</td>
<td>----------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| 0      | VSLunit Clock gating disable                     | R/W    | '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
|        |                                                  |        | '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)                  |
| 1      | VSHMunit clock gate disable                      | R/W    | '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
|        |                                                  |        | '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)                  |
| 2      | WAVMunit Clock Gate Disable                      | R/W    | WAVM units Clock Gating Disable Control:  
|        |                                                  |        | '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
|        |                                                  |        | '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)                  |
### Unit Level Clock Gating Control 6

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>09430h</td>
</tr>
</tbody>
</table>

#### UCGCTL6 - Unit Level Clock Gating Control 6

**Unit Level Clock Gating Disable bits**

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31</td>
<td>SPARE 3 clock gate disable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SPARE 3 unit Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>

| 30:28 | HDCunit clock gate disable |
|       | Access: R/W   |
|       | HDC units Clock Gating Disable Control: |
|       | '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) |
|       | '1': Clock Gating Disabled. (i.e., clocks are toggling, always) |

| 27       | MUCunit clock gate disable |
|          | Access: R/W   |
|          | MUC units Clock Gating Disable Control: |
|          | '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) |
|          | '1': Clock Gating Disabled. (i.e., clocks are toggling, always) |

| 26       | GACVunit clock gate disable |
|          | Access: R/W   |
|          | GACV units Clock Gating Disable Control: |
|          | '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) |
|          | '1': Clock Gating Disabled. (i.e., clocks are toggling, always) |
### UCGCTL6 - Unit Level Clock Gating Control 6

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>25 GACBunit clock gate disable</td>
<td>Access: R/W GACB units Clock Gating Disable Control: ’0’ : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) ’1’ : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
<tr>
<td>24 GAPSSunit clock gate disable</td>
<td>Access: R/W GAPSS units Clock Gating Disable Control: ’0’ : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) ’1’ : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
<tr>
<td>23 GAMTunit clock gate disable</td>
<td>Access: R/W GAMT units Clock Gating Disable Control: ’0’ : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) ’1’ : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
<tr>
<td>22 Reserved</td>
<td>Access: R/W Reserved</td>
<td></td>
</tr>
<tr>
<td>21 OASCREP</td>
<td>Access: R/W OASCREP units Clock Gating Disable Control: ’0’ : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) ’1’ : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
<tr>
<td>20 OAADDRunit clock gate disable bit</td>
<td>Access: R/W OAADDR units Clock Gating Disable Control: ’0’ : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) ’1’ : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
<tr>
<td>19 GACVunit clock gate disable</td>
<td>Access: R/W GACV units Clock Gating Disable Control: ’0’ : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) ’1’ : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
</tbody>
</table>
## UCGCTL6 - Unit Level Clock Gating Control 6

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
<th>Access</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>18</td>
<td><strong>BDMunit clock gate disable</strong>&lt;br&gt;Access: R/W</td>
<td></td>
<td>BDM units Clock Gating Disable Control:&lt;br&gt;&quot;0&quot;: Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)&lt;br&gt;&quot;1&quot;: Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>17</td>
<td><strong>GATSunit clock gate disable</strong>&lt;br&gt;Access: R/W</td>
<td></td>
<td>GATS units Clock Gating Disable Control:&lt;br&gt;&quot;0&quot;: Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)&lt;br&gt;&quot;1&quot;: Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>16</td>
<td><strong>OATREPunit clock gate disable</strong>&lt;br&gt;Access: R/W</td>
<td></td>
<td>OATREP units Clock Gating Disable Control:&lt;br&gt;&quot;0&quot;: Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)&lt;br&gt;&quot;1&quot;: Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>15</td>
<td><strong>STunit clock gate disable</strong>&lt;br&gt;Access: R/W</td>
<td></td>
<td>ST units Clock Gating Disable Control:&lt;br&gt;&quot;0&quot;: Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)&lt;br&gt;&quot;1&quot;: Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>14</td>
<td><strong>DEunit clock gate disable</strong>&lt;br&gt;Access: R/W</td>
<td></td>
<td>DE units Clock Gating Disable Control:&lt;br&gt;&quot;0&quot;: Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)&lt;br&gt;&quot;1&quot;: Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>13</td>
<td><strong>VIN(VID6) unit clock gate disable</strong>&lt;br&gt;Access: R/W</td>
<td></td>
<td>VIN(VID6) units Clock Gating Disable Control:&lt;br&gt;&quot;0&quot;: Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)&lt;br&gt;&quot;1&quot;: Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>
## UCGCTRL6 - Unit Level Clock Gating Control 6

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
<th>Access</th>
<th>Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>VIN(VID5) unit clock gate disable</td>
<td>R/W</td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>11</td>
<td>WVOP unit clock gate disable</td>
<td>R/W</td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>10</td>
<td>WUSB unit clock gate disable</td>
<td>R/W</td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>9</td>
<td>WSEC unit clock gate disable</td>
<td>R/W</td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>8</td>
<td>WRS unit clock gate disable</td>
<td>R/W</td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>7</td>
<td>WQRC unit clock gate disable</td>
<td>R/W</td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality) '1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>
### UCGCTL6 - Unit Level Clock Gating Control 6

<table>
<thead>
<tr>
<th></th>
<th>WMPC unit level clock gate disable</th>
<th></th>
<th>WINunit Clock gate disable</th>
<th></th>
<th>WHME unit clock gate disable</th>
<th></th>
<th>WAVMunit Clock Gate Disable</th>
<th></th>
<th>VSHMunit clock gate disable</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Access:</strong> R/W</td>
<td></td>
<td><strong>Access:</strong> R/W</td>
<td></td>
<td><strong>Access:</strong> R/W</td>
<td></td>
<td><strong>Access:</strong> R/W</td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td>6</td>
<td>WMPC units Clock Gating Disable Control:</td>
<td></td>
<td>WIN units Clock Gating Disable Control:</td>
<td></td>
<td>WHME units Clock Gating Disable Control:</td>
<td></td>
<td>WAVM units Clock Gating Disable Control:</td>
<td></td>
<td>VSHM units Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>Register</td>
<td>Description</td>
<td>Access</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UCGCTL6</td>
<td>Unit Level Clock Gating Control 6</td>
<td>R/W</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>VSLunit Clock gating disable</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>VSL units Clock Gating Disable Control:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### UCGCTL7 - Unit Level Clock Gating Control 7

- **Register Space:** MMIO: 0/2/0
- **Project:** CHV, BSW
- **Source:** PRM
- **Default Value:** 0x00000000
- **Size (in bits):** 32
- **Address:** 09438h

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:18</td>
<td><strong>RSVD</strong></td>
</tr>
<tr>
<td>17</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td>16</td>
<td></td>
<td><strong>cp_cg3ddis_hwm</strong></td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>HWM unit Clock Gating Disable (cp_cg3ddis_hwm)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>HWM unit Clock Gating Disable Control</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td></td>
<td><strong>cp_cg3ddis_hed</strong></td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>HED unit Clock Gating Disable (cp_cg3ddis_hed)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>HED unit Clock Gating Disable Control</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td><strong>cp_cg3ddis_hpp</strong></td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>HPP unit Clock Gating Disable (cp_cg3ddis_hpp)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>HPP unit Clock Gating Disable Control</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
<td><strong>cp_cg3ddis_hpr</strong></td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>HPR unit Clock Gating Disable (cp_cg3ddis_hpr)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>HPR unit Clock Gating Disable Control</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
</tbody>
</table>
## UCGCTL7 - Unit Level Clock Gating Control 7

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
<th>Access</th>
<th>Valuation</th>
</tr>
</thead>
<tbody>
<tr>
<td>cp_cg3ddis_hmc</td>
<td>HMC unit Clock Gating Disable (cp_cg3ddis_hmc )</td>
<td>R/W</td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td></td>
<td>HMC unit Clock Gating Disable Control</td>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cp_cg3ddis_hlf</td>
<td>HLF unit Clock Gating Disable (cp_cg3ddis_hlf )</td>
<td>R/W</td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td>HLF unit Clock Gating Disable Control</td>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>cp_cg3ddis_hmx</td>
<td>HMX unit Clock Gating Disable (cp_cg3ddis_hmx )</td>
<td>R/W</td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td>HMX unit Clock Gating Disable Control</td>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>cp_cg3ddis_vmm</td>
<td>VMM unit Clock Gating Disable (cp_cg3ddis_vmm )</td>
<td>R/W</td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td>VMM unit Clock Gating Disable Control</td>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>cp_cg3ddis_mpd</td>
<td>MPD unit Clock Gating Disable (cp_cg3ddis_mpd )</td>
<td>R/W</td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td>MPD unit Clock Gating Disable Control</td>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>cp_cg3ddis_mbd</td>
<td>MBD unit Clock Gating Disable (cp_cg3ddis_mbd )</td>
<td>R/W</td>
<td></td>
</tr>
</tbody>
</table>
## UCGCTL7 - Unit Level Clock Gating Control 7

<table>
<thead>
<tr>
<th>Access</th>
<th>Register</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>cp_cg3ddis_mlf</td>
<td>MLF unit Clock Gating Disable (cp_cg3ddis_mlf)</td>
</tr>
<tr>
<td>R/W</td>
<td></td>
<td>MLF unit Clock Gating Disable Control</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for</td>
</tr>
<tr>
<td></td>
<td></td>
<td>functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td></td>
<td>cp_cg3ddis_mmc</td>
<td>MMC unit Clock Gating Disable (cp_cg3ddis_mmc)</td>
</tr>
<tr>
<td>R/W</td>
<td></td>
<td>MMC unit Clock Gating Disable Control</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for</td>
</tr>
<tr>
<td></td>
<td></td>
<td>functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td></td>
<td>cp_cg3ddis_mmx</td>
<td>MMX unit Clock Gating Disable (cp_cg3ddis_mmx)</td>
</tr>
<tr>
<td>R/W</td>
<td></td>
<td>MMX unit Clock Gating Disable Control</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for</td>
</tr>
<tr>
<td></td>
<td></td>
<td>functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td></td>
<td>cp_cg3ddisvbsp</td>
<td>VBSP unit Clock Gating Disable (cp_cg3ddisvbsp)</td>
</tr>
<tr>
<td>R/W</td>
<td></td>
<td>VBSP unit Clock Gating Disable Control</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for</td>
</tr>
<tr>
<td></td>
<td></td>
<td>functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td></td>
<td>cp_cg3ddisvmpc</td>
<td>VMPC unit Clock Gating Disable (cp_cg3ddisvmpc)</td>
</tr>
<tr>
<td>R/W</td>
<td></td>
<td>VMPC unit Clock Gating Disable Control</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for</td>
</tr>
<tr>
<td></td>
<td></td>
<td>functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td></td>
<td>cp_cg3ddisvsec</td>
<td>VSEC unit Clock Gating Disable (cp_cg3ddisvsec)</td>
</tr>
<tr>
<td>R/W</td>
<td></td>
<td>VSEC unit Clock Gating Disable Control</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for</td>
</tr>
<tr>
<td></td>
<td></td>
<td>functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>UCGCTL7 - Unit Level Clock Gating Control 7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VSEC unit Clock Gating Disable Control</td>
<td></td>
<td></td>
</tr>
<tr>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>0</th>
<th>cp_cg3ddisjusb</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td>JUSB unit Clock Gating Disable (cp_cg3ddisjusb )</td>
<td></td>
</tr>
<tr>
<td>JUSB unit Clock Gating Disable Control</td>
<td></td>
</tr>
<tr>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
</tbody>
</table>
# Unit Level Clock Gating Control 7

## UCGCTL7 - Unit Level Clock Gating Control 7

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>09434h</td>
</tr>
</tbody>
</table>

### Unit Level Clock Gating Disable bits

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31</td>
<td><strong>wrcunit Clock Gating Disable</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>wrncunit Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>30</td>
<td></td>
<td><strong>mmcdunit Clock Gating Disable</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>mmcdunit Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>29</td>
<td></td>
<td><strong>bfceuunit Clock Gating Disable</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>bfcwunit Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>28</td>
<td></td>
<td><strong>ecpunit Clock Gating Disable</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ecpunit Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>Register</td>
<td>Description</td>
<td>Access</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------------------------------------------</td>
<td>--------</td>
</tr>
</tbody>
</table>
| vdlunit1 Clock Gating Disable | Access: R/W  
vdlnunit1 Clock Gating Disable Control:  
'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
'1': Clock Gating Disabled. (i.e., clocks are toggling, always) |        |                                      |
| vhmeunit Clock Gating Disable   | Access: R/W  
vhmeunit Clock Gating Disable Control:  
'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
'1': Clock Gating Disabled. (i.e., clocks are toggling, always) |        |                                      |
| vimeunit Clock Gating Disable   | Access: R/W  
vimeunit Clock Gating Disable Control:  
'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
'1': Clock Gating Disabled. (i.e., clocks are toggling, always) |        |                                      |
| vcreunit Clock Gating Disable   | Access: R/W  
vcreunit Clock Gating Disable Control:  
'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
'1': Clock Gating Disabled. (i.e., clocks are toggling, always) |        |                                      |
| vdxunit Clock Gating Disable   | Access: R/W  
vdxunit Clock Gating Disable Control:  
'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
'1': Clock Gating Disabled. (i.e., clocks are toggling, always) |        |                                      |
| mdcunit Clock Gating Disable   | Access: R/W  
mdcunit Clock Gating Disable Control:  
'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
'1': Clock Gating Disabled. (i.e., clocks are toggling, always) |        |                                      |
## UCGCTL7 - Unit Level Clock Gating Control 7

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pcctl7</td>
<td>Unit Level Clock Gating Disable</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Access</th>
<th>R/W</th>
</tr>
</thead>
</table>
| hpounit Clock Gating Disable | '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
|                  | '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)             |
| hrsunit Clock Gating Disable | '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
|                  | '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)             |
| ftunit Clock Gating Disable | '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
|                  | '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)             |
| fhunite Clock Gating Disable | '0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
|                  | '1' : Clock Gating Disabled. (i.e., clocks are toggling, always)             |

---

**Doc Ref # IHD-OS-CHV-BSW-Vol 2c-10.15**

1321
### UCGCTL7 - Unit Level Clock Gating Control 7

<table>
<thead>
<tr>
<th></th>
<th>hhiunit Clock Gating Disable</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Access:</strong></td>
<td><strong>R/W</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>hhiunit Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>mlfunit Clock Gating Disable</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Access:</strong></td>
<td><strong>R/W</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>mlfunit Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>mmcunit Clock Gating Disable</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Access:</strong></td>
<td><strong>R/W</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>mmcunit Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>mbdunit Clock Gating Disable</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Access:</strong></td>
<td><strong>R/W</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>mbdunit Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>mdpunit Clock Gating Disable</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Access:</strong></td>
<td><strong>R/W</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>mdpunit Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>mmxunit Clock Gating Disable</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Access:</strong></td>
<td><strong>R/W</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>mmxunit Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
</tbody>
</table>
## UCGCTL7 - Unit Level Clock Gating Control 7

<table>
<thead>
<tr>
<th>Register Access</th>
<th>Description</th>
</tr>
</thead>
</table>
| **9** hedunit Clock Gating Disable | Access: R/W  
 hedunit Clock Gating Disable Control:  
 '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
 '1': Clock Gating Disabled. (i.e., clocks are toggling, always) |
| **8** hlfunit Clock Gating Disable | Access: R/W  
 hlfunit Clock Gating Disable Control:  
 '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
 '1': Clock Gating Disabled. (i.e., clocks are toggling, always) |
| **7** hmcunit Clock Gating Disable | Access: R/W  
 hmcunit Clock Gating Disable Control:  
 '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
 '1': Clock Gating Disabled. (i.e., clocks are toggling, always) |
| **6** hmxunit Clock Gating Disable | Access: R/W  
 hmxunit Clock Gating Disable Control:  
 '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
 '1': Clock Gating Disabled. (i.e., clocks are toggling, always) |
| **5** hppunit Clock Gating Disable | Access: R/W  
 hppunit Clock Gating Disable Control:  
 '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
 '1': Clock Gating Disabled. (i.e., clocks are toggling, always) |
| **4** hprunit Clock Gating Disable | Access: R/W  
 hprunit Clock Gating Disable Control:  
 '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
 '1': Clock Gating Disabled. (i.e., clocks are toggling, always) |
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td><strong>hwmunit Clock Gating Disable</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>hwmunit Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td><strong>vmpcunit Clock Gating Disable</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>vmpcunit Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td><strong>vsecunit Clock Gating Disable</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>Access:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>vsecunit Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
</tbody>
</table>
## Unit Level Clock Gating Control 8

### UCGCTL8 - Unit Level Clock Gating Control 8

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31</td>
<td><strong>jusbunit Clock Gating Disable</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>jusbunit Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>30</td>
<td></td>
<td><strong>sfiunit Clock Gating Disable</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>sfiunit Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>29</td>
<td></td>
<td><strong>sfeunit Clock Gating Disable</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>sfeunit Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>28</td>
<td></td>
<td><strong>sfaunit Clock Gating Disable</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>sfaunit Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>
### UCGCTL8 - Unit Level Clock Gating Control 8

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sfounit</td>
<td>Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not</td>
</tr>
<tr>
<td></td>
<td>required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>sfxunit</td>
<td>Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not</td>
</tr>
<tr>
<td></td>
<td>required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>sfmunit</td>
<td>Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not</td>
</tr>
<tr>
<td></td>
<td>required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>vmmunit</td>
<td>Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not</td>
</tr>
<tr>
<td></td>
<td>required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>vrtunit</td>
<td>Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not</td>
</tr>
<tr>
<td></td>
<td>required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>ccunit</td>
<td>Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not</td>
</tr>
<tr>
<td></td>
<td>required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>
### UCGCT8 - Unit Level Clock Gating Control 8

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
<th>Access</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>gassunit</td>
<td>Clock Gating Disable</td>
<td>R/W</td>
<td>'0' : Clock Gating Enabled (i.e., clocks can be gated when they are not required to toggle for functionality) '1' : Clock Gating Disabled (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>gamdunit</td>
<td>Clock Gating Disable</td>
<td>R/W</td>
<td>'0' : Clock Gating Enabled (i.e., clocks can be gated when they are not required to toggle for functionality) '1' : Clock Gating Disabled (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>vdlunit1</td>
<td>Clock Gating Disable</td>
<td>R/W</td>
<td>'0' : Clock Gating Enabled (i.e., clocks can be gated when they are not required to toggle for functionality) '1' : Clock Gating Disabled (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>vhmeunit</td>
<td>Clock Gating Disable</td>
<td>R/W</td>
<td>'0' : Clock Gating Enabled (i.e., clocks can be gated when they are not required to toggle for functionality) '1' : Clock Gating Disabled (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>vcreunit</td>
<td>Clock Gating Disable</td>
<td>R/W</td>
<td>'0' : Clock Gating Enabled (i.e., clocks can be gated when they are not required to toggle for functionality) '1' : Clock Gating Disabled (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>hleunit</td>
<td>Clock Gating Disable</td>
<td>R/W</td>
<td>'0' : Clock Gating Enabled (i.e., clocks can be gated when they are not required to toggle for functionality) '1' : Clock Gating Disabled (i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>
## UCGCTL8 - Unit Level Clock Gating Control 8

<table>
<thead>
<tr>
<th></th>
<th>mbdunit Clock Gating Disable</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td>15</td>
<td>mbdunit Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>mmxunit Clock Gating Disable</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td>14</td>
<td>mmxunit Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>mpdunit Clock Gating Disable</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td>13</td>
<td>mpdunit Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>hedunit Clock Gating Disable</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td>12</td>
<td>hedunit Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>hlfunit Clock Gating Disable</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td>11</td>
<td>hlfunit Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>hmcunit Clock Gating Disable</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td>10</td>
<td>hmcunit Clock Gating Disable Control:</td>
</tr>
<tr>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>
## UCGCTL8 - Unit Level Clock Gating Control 8

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td><strong>hmxunit Clock Gating Disable</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>hmxunit Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td><strong>hppunit Clock Gating Disable</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>hppunit Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td><strong>hprunit Clock Gating Disable</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>hprunit Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td><strong>Reserved</strong></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td><strong>hwmunit Clock Gating Disable</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>hwmunit Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td><strong>mdcunit Clock Gating Disable</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>mdcunit Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td><strong>vmpcunit Clock Gating Disable</strong></td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>vmpcunit Clock Gating Disable Control:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>'1': Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
<td></td>
</tr>
</tbody>
</table>
### UCGCTL8 - Unit Level Clock Gating Control 8

<table>
<thead>
<tr>
<th>Address</th>
<th>Register Description</th>
<th>Access</th>
<th>Control</th>
</tr>
</thead>
</table>
| 2       | sfmunit Clock Gating Disable ebb | R/W | '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
          |           |       | '1': Clock Gating Disabled. (i.e., clocks are toggling, always) |
| 1       | sfaunit Clock Gating Disable ebb   | R/W | '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
          |           |       | '1': Clock Gating Disabled. (i.e., clocks are toggling, always) |
| 0       | sfeunit Clock Gating Disable ebb   | R/W | '0': Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)  
          |           |       | '1': Clock Gating Disabled. (i.e., clocks are toggling, always) |
## Unit Level Clock Gating Control 9

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:4</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>RO</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td><strong>vbspunit Clock Gating Disable</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>R/W</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>vbspunit Clock Gating Disable Control:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td><strong>vmmunit Clock Gating Disable</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>R/W</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>vmmunit Clock Gating Disable Control:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td><strong>AVSunit Clock Gating Disable</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>R/W</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>AVSunit Clock Gating Disable Control:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td><strong>daprssunit Clock Gating Disable</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>R/W</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>daprssunit Clock Gating Disable Control:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0' : Clock Gating Enabled. (i.e., clocks can be gated when they are not required to toggle for functionality)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'1' : Clock Gating Disabled. (i.e., clocks are toggling, always)</td>
</tr>
</tbody>
</table>
# URB Context Offset

<table>
<thead>
<tr>
<th>URB_CXT_OFFSET - URB Context Offset</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> RenderCS</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00009AC0</td>
</tr>
<tr>
<td><strong>Access:</strong> Read/32 bit Write Only</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 021B8h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:6</td>
<td><strong>URB Offset</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 26Bh</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This field indicates the offset (64bytes granular) in to the logical rendering context to which URB contents are save/restored when enabled. This field register must not be written directly (via MMIO) unless the Command Streamer is completely idle (i.e., the Ring Buffer is empty and the pipeline is idle) and RC6 is disabled. One way to program this register is via Load Register Immediate command in the ring buffer as part of initialization sequence.</td>
</tr>
<tr>
<td>5:0</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
</tbody>
</table>
### Valid Bit Vector 0 for CVS

<table>
<thead>
<tr>
<th><strong>Register Space:</strong></th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Project:</strong></td>
<td>CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong></td>
<td>PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong></td>
<td>0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong></td>
<td>32</td>
</tr>
<tr>
<td><strong>Address:</strong></td>
<td>04C00h</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th><strong>DWord</strong></th>
<th><strong>Bit</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Valid Bit Vector 0 for CVS</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Valid Bits per Entry.</strong></td>
</tr>
</tbody>
</table>

**Valid Bit Vector 0 for CVS**

---

*Valid Bit Vector 0 for CVS*
Valid Bit Vector 0 for L3

<table>
<thead>
<tr>
<th>L3TLB_VLD_0 - Valid Bit Vector 0 for L3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 04D00h</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Valid Bit Vector 0 for L3</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Bits per Entry.</td>
</tr>
</tbody>
</table>
Valid Bit Vector 0 for MFX

<table>
<thead>
<tr>
<th>MFXTLB_VLD_0 - Valid Bit Vector 0 for MFX</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 04BA0h</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Valid Bit Vector 0 for MFX</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Bits per Entry.</td>
</tr>
</tbody>
</table>
Valid Bit Vector 0 for MFX SL1

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>Valid Bit Vector 0 for MFX SL1</td>
</tr>
</tbody>
</table>

Default Value: $00000000h$

Access: RO

Valid Bits per Entry.
Valid Bit Vector 0 for MTTLB

<table>
<thead>
<tr>
<th>MTTLB_VLD0 - Valid Bit Vector 0 for MTTLB</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> RenderCS</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Trusted Type:</strong> 1</td>
</tr>
<tr>
<td><strong>Address:</strong> 04780h-04783h</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Valid bits per entry</strong></td>
</tr>
</tbody>
</table>
## Valid Bit Vector 0 for MTVICTLB

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>Valid bits per entry</td>
</tr>
</tbody>
</table>
Valid Bit Vector 0 for RCC

<table>
<thead>
<tr>
<th>RCCTLB_VLD_0 - Valid Bit Vector 0 for RCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 04DA0h</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>Valid Bit Vector 0 for RCC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Bits per Entry.</td>
</tr>
</tbody>
</table>
**Valid Bit Vector 0 for RCCTLB**

<table>
<thead>
<tr>
<th>RCCTLB_VLD0 - Valid Bit Vector 0 for RCCTLB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: RenderCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Access: RO</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Trusted Type: 1</td>
</tr>
<tr>
<td>Address: 04790h-04793h</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Valid bits per entry</strong></td>
</tr>
</tbody>
</table>
Valid Bit Vector 0 for RCZTLB

<table>
<thead>
<tr>
<th>RCZTLB_VLD0 - Valid Bit Vector 0 for RCZTLB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: RenderCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Access: RO</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Trusted Type: 1</td>
</tr>
<tr>
<td>Address: 04798h-0479Bh</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>Valid bits per entry</td>
</tr>
</tbody>
</table>
Valid Bit Vector 0 for TLBPEND registers

<table>
<thead>
<tr>
<th>TLBPEND_VLD0 - Valid Bit Vector 0 for TLBPEND registers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: RenderCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Access: R/W</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Trusted Type: 1</td>
</tr>
<tr>
<td>Address: 04700h-04703h</td>
</tr>
<tr>
<td>This register contains the valid bits for entries 0-31 of TLBPEND structure (Cycles pending TLB translation).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>Valid bits per entry</td>
</tr>
</tbody>
</table>
## Valid Bit Vector 0 for VEBX

<table>
<thead>
<tr>
<th><strong>VEBXTLB_VLD_0 - Valid Bit Vector 0 for VEBX</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong></td>
<td>MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong></td>
<td>CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong></td>
<td>PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong></td>
<td>0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong></td>
<td>32</td>
</tr>
<tr>
<td><strong>Address:</strong></td>
<td>04B20h</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
</table>
| 0 | 31:0 | **Valid Bit Vector 0 for VEBX**

| Access: | RO |
| Default Value: | 00000000h |

Valid Bits per Entry.
## Valid Bit Vector 0 for WIDI

<table>
<thead>
<tr>
<th>BWDTLB_VLD_0 - Valid Bit Vector 0 for WIDI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 04DC0h</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>Valid Bit Vector 0 for WIDI</td>
</tr>
</tbody>
</table>

- Default Value: 00000000h
- Access: RO

Valid Bits per Entry.
## Valid Bit Vector 0 for Z

<table>
<thead>
<tr>
<th><strong>ZTLB_VLD_0 - Valid Bit Vector 0 for Z</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 04B34h</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th><strong>DWord</strong></th>
<th><strong>Bit</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Valid Bit Vector 0 for Z</strong></td>
</tr>
</tbody>
</table>

**Default Value:** 00000000h

**Access:** RO

Valid Bits per Entry.
## Valid Bit Vector 1 for CVS

<table>
<thead>
<tr>
<th>CVSTLB_VLD_1 - Valid Bit Vector 1 for CVS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 04C04h</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Valid Bit Vector 1 for CVS</strong></td>
</tr>
</tbody>
</table>

|  | Default Value: | 00000000h |
|  | Access: | RO |

Valid Bits per Entry.
# Valid Bit Vector 1 for L3

<table>
<thead>
<tr>
<th><strong>L3TLB_VLD_1 - Valid Bit Vector 1 for L3</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 04D04h</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th><strong>DWord</strong></th>
<th><strong>Bit</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Valid Bit Vector 1 for L3</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Bits per Entry.</td>
</tr>
</tbody>
</table>
**Valid Bit Vector 1 for MFX**

<table>
<thead>
<tr>
<th>MFXTLB_VLD_1 - Valid Bit Vector 1 for MFX</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 04BA4h</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Valid Bit Vector 1 for MFX</strong></td>
</tr>
</tbody>
</table>

- **Default Value:** 00000000h
- **Access:** RO
- **Valid Bits per Entry.**
Valid Bit Vector 1 for MFX SL1

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>Valid Bit Vector 1 for MFX SL1</td>
<td>00000000h</td>
<td>RO</td>
</tr>
</tbody>
</table>

This register contains the valid bits for entries 0-31 of MFX SL1 TLB.
Valid Bit Vector 1 for MTTLB

<table>
<thead>
<tr>
<th>MTTLB_VLD1 - Valid Bit Vector 1 for MTTLB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: RenderCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Access: RO</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Trusted Type: 1</td>
</tr>
<tr>
<td>Address: 04784h-04787h</td>
</tr>
</tbody>
</table>

This register contains the valid bits for entries 0-31 of MTTLB (Texture and constant cache TLBVertex Fetch, Instruction Cache, and Command Streamer TLB).

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>Valid bits per entry</td>
</tr>
</tbody>
</table>
## Valid Bit Vector 1 for MTVICTLB

### MTVICTLB_VLD1 - Valid Bit Vector 1 for MTVICTLB

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>RenderCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>RO</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Trusted Type:</td>
<td>1</td>
</tr>
<tr>
<td>Address:</td>
<td>0478Ch-0478Fh</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>Valid bits per entry</td>
</tr>
</tbody>
</table>
## Valid Bit Vector 1 for RCC

<table>
<thead>
<tr>
<th>RCCTLB_VLD_1 - Valid Bit Vector 1 for RCC</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 04DA4h</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Valid Bit Vector 1 for RCC</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Bits per Entry.</td>
</tr>
</tbody>
</table>
Valid Bit Vector 1 for RCCTLB

<table>
<thead>
<tr>
<th>RCCTLB_VLD1 - Valid Bit Vector 1 for RCCTLB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: RenderCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Access: RO</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Trusted Type: 1</td>
</tr>
<tr>
<td>Address: 04794h-04797h</td>
</tr>
</tbody>
</table>

This register is reserved for future RCC TLB extension.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Reserved</strong></td>
</tr>
</tbody>
</table>

Format: MBZ
## Valid Bit Vector 1 for RCZTLB

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Valid bits per entry</strong></td>
</tr>
</tbody>
</table>

This register contains the valid bits for entries 0-31 of RCZTLB (Render Cache for Z (Depth), Hi Z, and Stencil TLB).
Valid Bit Vector 1 for TLBPEND registers

<table>
<thead>
<tr>
<th>TLBPEND_VLD1 - Valid Bit Vector 1 for TLBPEND registers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: RenderCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Access: R/W</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Trusted Type: 1</td>
</tr>
<tr>
<td>Address: 04704h-04707h</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>Valid bits per entry</td>
</tr>
</tbody>
</table>
Valid Bit Vector 1 for VEBX

### VEBXTLB_VLD_1 - Valid Bit Vector 1 for VEBX

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>04B24h</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Valid Bit Vector 1 for VEBX</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Bits per Entry.</td>
</tr>
</tbody>
</table>
Valid Bit Vector 1 for WIDI

<table>
<thead>
<tr>
<th>BWDTLB_VLD_1 - Valid Bit Vector 1 for WIDI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 04DC4h</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Valid Bit Vector 1 for WIDI</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Bits per Entry.</td>
</tr>
</tbody>
</table>
# Valid Bit Vector 1 for Z

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ZTLB_VLD_1</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Register Space</td>
<td>MMIO: 0/2/0</td>
<td></td>
</tr>
<tr>
<td>Project</td>
<td>CHV, BSW</td>
<td></td>
</tr>
<tr>
<td>Source</td>
<td>PRM</td>
<td></td>
</tr>
<tr>
<td>Default Value</td>
<td>0x00000000</td>
<td></td>
</tr>
<tr>
<td>Size (in bits)</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>Address</td>
<td>04B38h</td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Valid Bit Vector 1 for Z</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Bits per Entry.</td>
</tr>
</tbody>
</table>
## Valid Bit Vector 2 for CVS

<table>
<thead>
<tr>
<th>CVSTLB_VLD_2 - Valid Bit Vector 2 for CVS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space:</td>
</tr>
<tr>
<td>Project:</td>
</tr>
<tr>
<td>Source:</td>
</tr>
<tr>
<td>Default Value:</td>
</tr>
<tr>
<td>Size (in bits):</td>
</tr>
<tr>
<td>Address:</td>
</tr>
</tbody>
</table>

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

### DWord | Bit | Description
---|---|---
0 | 31:0 | Valid Bit Vector 2 for CVS
| | | Default Value: 00000000h |
| | | Access: RO |
| | | Valid Bits per Entry.
Valid Bit Vector 2 for GAB

<table>
<thead>
<tr>
<th>BWDTLB_VLD_3 - Valid Bit Vector 2 for GAB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x0000000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 04DCCh</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>Valid Bit Vector 3 for GAB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Bits per Entry.</td>
</tr>
</tbody>
</table>
Valid Bit Vector 2 for L3

<table>
<thead>
<tr>
<th>L3TLB_VLD_2 - Valid Bit Vector 2 for L3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x0000000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 04D08h</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>Valid Bit Vector 2 for L3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
</tbody>
</table>

Valid Bits per Entry.
## Valid Bit Vector 2 for MFX

**MFXTLB_VLD_2 - Valid Bit Vector 2 for MFX**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>04BA8h</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Valid Bit Vector 2 for MFX</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Bits per Entry.</td>
</tr>
</tbody>
</table>


Valid Bit Vector 2 for MFX SL1

<table>
<thead>
<tr>
<th>MFXTLB_VLD_SL1_2 - Valid Bit Vector 2 for MFX SL1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 04BC8h</td>
</tr>
</tbody>
</table>

This register contains the valid bits for entries 0-31 of MFX SL1 TLB.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>Valid Bit Vector 2 for MFX SL1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Bits per Entry.</td>
</tr>
</tbody>
</table>

Valid Bit Vector 2 for MFX SL1
## Valid Bit Vector 2 for RCC

<table>
<thead>
<tr>
<th>RCCTLB_VLD_2 - Valid Bit Vector 2 for RCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 04DA8h</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Valid Bit Vector 2 for RCC</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Bits per Entry.</td>
</tr>
</tbody>
</table>
# Valid Bit Vector 2 for Z

<table>
<thead>
<tr>
<th>ZTLB_VLD_2 - Valid Bit Vector 2 for Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 04B3Ch</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Valid Bit Vector 2 for Z</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Bits per Entry.</td>
</tr>
</tbody>
</table>
## Valid Bit Vector 3 for CVS

<table>
<thead>
<tr>
<th>CVSTLB_VLD_3 - Valid Bit Vector 3 for CVS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 04C0Ch</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>Valid Bit Vector 3 for CVS</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Bits per Entry.</td>
</tr>
</tbody>
</table>

---

*Valid Bit Vector 3 for CVS*
Valid Bit Vector 3 for L3

<table>
<thead>
<tr>
<th>L3TLB_VLD_3 - Valid Bit Vector 3 for L3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 04D0Ch</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Valid Bit Vector 3 for L3</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> RO</td>
</tr>
</tbody>
</table>

Valid Bits per Entry.
Valid Bit Vector 3 for MFX

<table>
<thead>
<tr>
<th>MFXTLB_VLD_3 - Valid Bit Vector 3 for MFX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 04BACh</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>Valid Bit Vector 3 for MFX</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Bits per Entry.</td>
</tr>
</tbody>
</table>
Valid Bit Vector 3 for MFX SL1

<table>
<thead>
<tr>
<th>MFXTLB_VLD_SL1_3 - Valid Bit Vector 3 for MFX SL1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 04BCCh</td>
</tr>
</tbody>
</table>

This register contains the valid bits for entries 0-31 of MFX SL1 TLB.

### DWord | Bit | Description |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>Valid Bit Vector 3 for MFX SL1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Default Value:</th>
<th>00000000h</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access:</td>
<td>RO</td>
</tr>
<tr>
<td>Valid Bits per Entry.</td>
<td></td>
</tr>
</tbody>
</table>
Valid Bit Vector 3 for RCC

<table>
<thead>
<tr>
<th>RCCTLB_VLD_3 - Valid Bit Vector 3 for RCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 04DACh</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>Valid Bit Vector 3 for RCC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Bits per Entry.</td>
</tr>
</tbody>
</table>
Valid Bit Vector 3 for Z

<table>
<thead>
<tr>
<th>ZTLB_VLD_3 - Valid Bit Vector 3 for Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 04B40h</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>Valid Bit Vector 3 for Z</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Bits per Entry.</td>
</tr>
</tbody>
</table>
## Valid Bit Vector 4 for L3

<table>
<thead>
<tr>
<th>L3TLB_VLD_4 - Valid Bit Vector 4 for L3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 04D10h</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Valid Bit Vector 4 for L3</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Bits per Entry.</td>
</tr>
</tbody>
</table>
## Valid Bit Vector 4 for MFX

<table>
<thead>
<tr>
<th>MFXTLB_VLD_4 - Valid Bit Vector 4 for MFX</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 04BB0h</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>Valid Bit Vector 4 for MFX</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Bits per Entry.</td>
</tr>
</tbody>
</table>
Valid Bit Vector 4 for MFX SL1

<table>
<thead>
<tr>
<th>MFXTLB_VLD_SL1_4 - Valid Bit Vector 4 for MFX SL1</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 04BD0h</td>
</tr>
</tbody>
</table>

This register contains the valid bits for entries 0-31 of MFX SL1 TLB.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>Valid Bit Vector 4 for MFX SL1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Bits per Entry.</td>
</tr>
</tbody>
</table>
# Valid Bit Vector 4 for RCC

<table>
<thead>
<tr>
<th>RCCTLB_VLD_4 - Valid Bit Vector 4 for RCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 04DB0h</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Valid Bit Vector 4 for RCC</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Bits per Entry.</td>
</tr>
</tbody>
</table>
### Valid Bit Vector 4 for Z

**ZTLB_VLD_4 - Valid Bit Vector 4 for Z**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>04B44h</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Valid Bit Vector 4 for Z</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Bits per Entry.</td>
</tr>
</tbody>
</table>
# Valid Bit Vector 5 for L3

<table>
<thead>
<tr>
<th>L3TLB_VLD_5 - Valid Bit Vector 5 for L3</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Address:</strong> 04D14h</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Valid Bit Vector 5 for L3</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Valid Bits per Entry.</strong></td>
</tr>
</tbody>
</table>
## Valid Bit Vector 5 for MFX

<table>
<thead>
<tr>
<th>MFXTLB_VLD_5</th>
<th>Valid Bit Vector 5 for MFX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space:</td>
<td>MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>04BB4h</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Valid Bit Vector 5 for MFX</strong></td>
</tr>
</tbody>
</table>

| | Default Value: | 00000000h |
| | Access: | RO |

Valid Bits per Entry.
## Valid Bit Vector 5 for MFX SL1

<table>
<thead>
<tr>
<th><strong>MFXTLB_VLD_SL1_5 - Valid Bit Vector 5 for MFX SL1</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
</tbody>
</table>

Address: 04BD4h

This register contains the valid bits for entries 0-31 of MFX SL1 TLB.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Valid Bit Vector 5 for MFX SL1</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Bits per Entry.</td>
</tr>
</tbody>
</table>
## Valid Bit Vector 5 for RCC

<table>
<thead>
<tr>
<th><strong>RCCTLB_VLD_5 - Valid Bit Vector 5 for RCC</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x000000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 04DB4h</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>Valid Bit Vector 5 for RCC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Bits per Entry.</td>
</tr>
</tbody>
</table>
# ZTLB_VLD_5 - Valid Bit Vector 5 for Z

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Valid Bit Vector 5 for Z</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Valid Bits per Entry.</strong></td>
</tr>
</tbody>
</table>
Valid Bit Vector 6 for L3

**L3TLB_VLD_6 - Valid Bit Vector 6 for L3**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>04D18h</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Valid Bit Vector 6 for L3</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Bits per Entry.</td>
</tr>
</tbody>
</table>
Valid Bit Vector 6 for MFX

<table>
<thead>
<tr>
<th>MFXTLB_VLD_6 - Valid Bit Vector 6 for MFX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space:</td>
</tr>
<tr>
<td>MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project:</td>
</tr>
<tr>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
</tr>
<tr>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
</tr>
<tr>
<td>0x000000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
</tr>
<tr>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
</tr>
<tr>
<td>04BB8h</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Valid Bit Vector 6 for MFX</strong></td>
</tr>
</tbody>
</table>

Default Value: 00000000h
Access: RO
Valid Bits per Entry.
### Valid Bit Vector 6 for MFX SL1

#### MFXTLB_VLD_SL1_6 - Valid Bit Vector 6 for MFX SL1

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>04BD8h</td>
</tr>
</tbody>
</table>

This register contains the valid bits for entries 0-31 of MFX SL1 TLB.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Valid Bit Vector 6 for MFX SL1</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Bits per Entry.</td>
</tr>
</tbody>
</table>
Valid Bit Vector 6 for RCC

**RCCTLB_VLD_6 - Valid Bit Vector 6 for RCC**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>04DB8h</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Valid Bit Vector 6 for RCC</strong></td>
</tr>
</tbody>
</table>

| Default Value: | 00000000h |
| Access:        | RO        |

Valid Bits per Entry.
## Valid Bit Vector 6 for Z

<table>
<thead>
<tr>
<th>ZTLB_VLD_6 - Valid Bit Vector 6 for Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 04B4Ch</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Valid Bit Vector 6 for Z</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Bits per Entry.</td>
</tr>
</tbody>
</table>


## Valid Bit Vector 7 for L3

<table>
<thead>
<tr>
<th>L3TLB_VLD_7 - Valid Bit Vector 7 for L3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0000000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 04D1Ch</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>Valid Bit Vector 7 for L3</td>
</tr>
</tbody>
</table>

- Default Value: 00000000h
- Access: RO

Valid Bits per Entry.
## Valid Bit Vector 7 for MFX

<table>
<thead>
<tr>
<th>MFXTLB_VLD_7 - Valid Bit Vector 7 for MFX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space:</td>
</tr>
<tr>
<td>Project:</td>
</tr>
<tr>
<td>Source:</td>
</tr>
<tr>
<td>Default Value:</td>
</tr>
<tr>
<td>Size (in bits):</td>
</tr>
<tr>
<td>Address:</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Valid Bit Vector 7 for MFX</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Bits per Entry</td>
</tr>
</tbody>
</table>
## Valid Bit Vector 7 for MFX SL1

**MFXTLB_VLD_SL1_7 - Valid Bit Vector 7 for MFX SL1**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>04BDCh</td>
</tr>
</tbody>
</table>

This register contains the valid bits for entries 0-31 of MFX SL1 TLB.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value:</th>
<th>Access:</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Valid Bit Vector 7 for MFX SL1</strong></td>
<td>00000000h</td>
<td>RO</td>
</tr>
</tbody>
</table>

Valid Bits per Entry.
Valid Bit Vector 7 for RCC

<table>
<thead>
<tr>
<th>RCCTLB_VLD_7 - Valid Bit Vector 7 for RCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space:</td>
</tr>
<tr>
<td>MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project:</td>
</tr>
<tr>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
</tr>
<tr>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
</tr>
<tr>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
</tr>
<tr>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
</tr>
<tr>
<td>04DBCh</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Valid Bit Vector 7 for RCC</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Bits per Entry.</td>
</tr>
</tbody>
</table>
Valid Bit Vector 7 for Z

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>04B50h</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Valid Bit Vector 7 for Z</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Bits per Entry.</td>
</tr>
</tbody>
</table>
## Valid Bit Vector 8 for L3

### L3TLB_VLD_8 - Valid Bit Vector 8 for L3

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>04D20h</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Valid Bit Vector 8 for L3</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Bits per Entry.</td>
</tr>
</tbody>
</table>
## Valid Bit Vector 8 for Z

<table>
<thead>
<tr>
<th>ZTLB_VLD_8 - Valid Bit Vector 8 for Z</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x0000000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 04B54h</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th><strong>DWord</strong></th>
<th><strong>Bit</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Valid Bit Vector 8 for Z</strong></td>
</tr>
</tbody>
</table>

- **Default Value:** 00000000h
- **Access:** RO
- **Valid Bits per Entry.**
## Valid Bit Vector 9 for L3

### L3TLB_VLD_9 - Valid Bit Vector 9 for L3

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>04D24h</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Valid Bit Vector 9 for L3</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Bits per Entry.</td>
</tr>
</tbody>
</table>
## Valid Bit Vector 9 for Z

<table>
<thead>
<tr>
<th><strong>ZTLB_VLD_9 - Valid Bit Vector 9 for Z</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space:</td>
<td>MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>04B58h</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Valid Bit Vector 9 for Z</strong></td>
</tr>
</tbody>
</table>

- Default Value: 00000000h
- Access: RO
- Valid Bits per Entry.
# Valid Bit Vector 10 for L3

<table>
<thead>
<tr>
<th>L3TLB_VLD_10 - Valid Bit Vector 10 for L3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 04D28h</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Valid Bit Vector 10 for L3</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Bits per Entry.</td>
</tr>
</tbody>
</table>
Valid Bit Vector 10 for Z

<table>
<thead>
<tr>
<th>ZTLB_VLD_10 - Valid Bit Vector 10 for Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 04B5Ch</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Valid Bit Vector 10 for Z</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Bits per Entry.</td>
</tr>
</tbody>
</table>

*Valid Bit Vector 10 for Z*
Valid Bit Vector 11 for L3

<table>
<thead>
<tr>
<th>L3TLB_VLD_11 - Valid Bit Vector 11 for L3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x000000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 04D2Ch</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Valid Bit Vector 11 for L3</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Bits per Entry.</td>
</tr>
</tbody>
</table>
Valid Bit Vector 11 for Z

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>04B60h</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>Valid Bit Vector 11 for Z</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Bits per Entry.</td>
</tr>
</tbody>
</table>
## Valid Bit Vector 12 for L3

<table>
<thead>
<tr>
<th><strong>L3TLB_VLD_12 - Valid Bit Vector 12 for L3</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 04D30h</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th><strong>DWord</strong></th>
<th><strong>Bit</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Valid Bit Vector 12 for L3</strong></td>
</tr>
</tbody>
</table>

|  |  | **Default Value:** 00000000h |
|  |  | **Access:** RO |

Valid Bits per Entry.
## Valid Bit Vector 12 for Z

### ZTLB_VLD_12 - Valid Bit Vector 12 for Z

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space</td>
<td>MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits)</td>
<td>32</td>
</tr>
<tr>
<td>Address</td>
<td>04B64h</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>Valid Bit Vector 12 for Z</td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Bits per Entry.</td>
<td></td>
</tr>
</tbody>
</table>


Valid Bit Vector 13 for L3

<table>
<thead>
<tr>
<th>L3TLB_VLD_13 - Valid Bit Vector 13 for L3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space:</td>
</tr>
<tr>
<td>MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project:</td>
</tr>
<tr>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
</tr>
<tr>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
</tr>
<tr>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
</tr>
<tr>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
</tr>
<tr>
<td>04D34h</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Valid Bit Vector 13 for L3</strong></td>
</tr>
</tbody>
</table>

- **Default Value:** 00000000h
- **Access:** RO
- Valid Bits per Entry.
## Valid Bit Vector 13 for Z

### ZTLB_VLD_13 - Valid Bit Vector 13 for Z

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>04B68h</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Valid Bit Vector 13 for Z</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Bits per Entry.</td>
</tr>
</tbody>
</table>
## Valid Bit Vector 14 for L3

<table>
<thead>
<tr>
<th>L3TLB_VLD_14 - Valid Bit Vector 14 for L3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 04D38h</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Valid Bit Vector 14 for L3</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
</tbody>
</table>

Valid Bits per Entry.
## Valid Bit Vector 14 for Z

<table>
<thead>
<tr>
<th>ZTLB_VLD_14 - Valid Bit Vector 14 for Z</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x0000000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 04B6Ch</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Valid Bit Vector 14 for Z</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Valid Bits per Entry.</strong></td>
</tr>
</tbody>
</table>
Valid Bit Vector 15 for L3

<table>
<thead>
<tr>
<th>L3TLB_VLD_15 - Valid Bit Vector 15 for L3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 04D3Ch</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Valid Bit Vector 15 for L3</strong></td>
</tr>
</tbody>
</table>

- **Default Value:** 00000000h
- **Access:** RO
- **Valid Bits per Entry.**
## Valid Bit Vector 15 for Z

### ZTLB_VLD_15 - Valid Bit Vector 15 for Z

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
</tbody>
</table>

**Address:** 04B70h

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Valid Bit Vector 15 for Z</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Bits per Entry.</td>
</tr>
</tbody>
</table>
Valid Bit Vector 16 for L3

<table>
<thead>
<tr>
<th>L3TLB_VLD_16 - Valid Bit Vector 16 for L3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 04D40h</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Valid Bit Vector 16 for L3</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Bits per Entry.</td>
</tr>
</tbody>
</table>

Valid Bit Vector 16 for L3
## Valid Bit Vector 17 for L3

### L3TLB_VLD_17 - Valid Bit Vector 17 for L3

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>04D44h</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Valid Bit Vector 17 for L3</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Bits per Entry.</td>
</tr>
</tbody>
</table>
Valid Bit Vector 18 for L3

<table>
<thead>
<tr>
<th>L3TLB_VLD_18 - Valid Bit Vector 18 for L3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 04D48h</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Valid Bit Vector 18 for L3</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Bits per Entry.</td>
</tr>
</tbody>
</table>
## Valid Bit Vector 19 for L3

### L3TLB_VLD_19 - Valid Bit Vector 19 for L3

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
</tbody>
</table>

Address: 04D4Ch

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>Valid Bit Vector 19 for L3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Bits per Entry.</td>
</tr>
</tbody>
</table>
Valid Bit Vector 20 for L3

<table>
<thead>
<tr>
<th>L3TLB_VLD_20 - Valid Bit Vector 20 for L3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 04D50h</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Valid Bit Vector 20 for L3</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Bits per Entry.</td>
</tr>
</tbody>
</table>
Valid Bit Vector 21 for L3

**L3TLB_VLD_21 - Valid Bit Vector 21 for L3**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
</tbody>
</table>

Address: 04D54h

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Valid Bit Vector 21 for L3</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Bits per Entry.</td>
</tr>
</tbody>
</table>
## Valid Bit Vector 22 for L3

<table>
<thead>
<tr>
<th>L3TLB_VLD_22 - Valid Bit Vector 22 for L3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 04D58h</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Valid Bit Vector 22 for L3</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid Bits per Entry.</td>
</tr>
</tbody>
</table>
Valid Bit Vector 23 for L3

<table>
<thead>
<tr>
<th>L3TLB_VLD_23 - Valid Bit Vector 23 for L3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x000000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 04D5Ch</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>Valid Bit Vector 23 for L3</td>
</tr>
</tbody>
</table>

  | Default Value: 00000000h |
  | Access: RO               |

Valid Bits per Entry.
Valid Bit Vector for VLF

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>Valid Bit Vector for VLF</td>
</tr>
</tbody>
</table>

Default Value: 00000000h
Access: RO
Valid Bits per Entry.

This register contains the valid bits for entries 0-31 of VLFTLB.
Valid Bit Vector for VLFSL1

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>Valid Bit Vector for VLFSL1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
</tbody>
</table>

Valid Bits per Entry.
### VC - VC

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:1</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td><strong>Spare_bit</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Placeholder for SKU related fusing.</td>
</tr>
</tbody>
</table>

Register Space: PCI: 0/2/0
Project: CHV, BSW
Source: PRM
Default Value: 0x00000000
Size (in bits): 32
Address: 000B4h

Vendor Capabilities. Any SKU related fuses would be added here.
## VCES Idle Switch Delay

<table>
<thead>
<tr>
<th><strong>VECS_IDLEDLY - VCES Idle Switch Delay</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> VideoEnhancementCS</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 1A23Ch</td>
</tr>
</tbody>
</table>

The IDLEDLY register contains an Idle Delay field which specifies the minimum number of microseconds allowed for command streamer to wait before a context is switched out leading to IDLE state in execlists mode, i.e. following this context switch there is no active element available in HW to execute.

A default value of 0, means that by default, there is no restriction to wait on a context switch leading to IDLE. This register has no significance when execlists are not enabled.

<table>
<thead>
<tr>
<th><strong>DWord</strong></th>
<th><strong>Bit</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:21</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: All</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
<tr>
<td>20:0</td>
<td></td>
<td><strong>IDLE Delay</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: All</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: U21</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Minimum number of micro-seconds allowed.</td>
</tr>
</tbody>
</table>
**Vendor Capability ID**

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>VERSION</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 01h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VS: Identifies this as the first revision of the CAPID register definition.</td>
</tr>
<tr>
<td>23:16</td>
<td></td>
<td><strong>LENGTH</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 07h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LEN: This field has the value 07h to indicate structure length (8 bytes)</td>
</tr>
<tr>
<td>15:8</td>
<td></td>
<td><strong>NEXT_CAPABILITY_POINTER</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W Once</td>
</tr>
<tr>
<td></td>
<td></td>
<td>00 indicates capability list ends here.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This register should be programmed by BIOS during boot-up. Once written, this register becomes Read_Only. This register can only be cleared by a Reset. Write-once allowing the capability list to be changed.</td>
</tr>
<tr>
<td>7:0</td>
<td></td>
<td><strong>CAPABILITY_ID_CID</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 09h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Identifies this as a vendor dependent capability pointers.</td>
</tr>
</tbody>
</table>
VCS_PREEMPTION_HINT

<table>
<thead>
<tr>
<th><strong>VCS_PREEMPTION_HINT - VCS_PREEMPTION_HINT</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> VideoCS</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 124BCh</td>
</tr>
<tr>
<td><strong>Valid Projects:</strong> CHV, BSW</td>
</tr>
</tbody>
</table>

This register contains the Head pointer offset in to the Ring Buffer or the Dword aligned Graphics address in to the Batch Buffer corresponding to either MI_ARB_CHECK called Preemption Hint Address. When Preemption Hint Address is enabled, VCS will honor UHPTR only on parsing MI_ARB_CHK at Preemption Hint Address.

This register contains the Head pointer offset into the Ring Buffer or the Dword aligned Graphics address into the Batch Buffer corresponding to the below listed commands in execute list mode of operation

- MI_ARB_CHECK
- MI_WAIT_FOR_EVENT
- MI_SEMAPHORE_WAIT

### Programming Notes

**Programming Restriction:** This register should NEVER be programmed in functional mode, this should be used only in validation mode to achieve deterministic behavior of UHPTR being sampled by a given MI_ARB_CHECK in command stream. Programmer has to ensure that VCS Preemption Hint register gets programmed before UHPTR is programmed and well before VCS crosses the corresponding execution point. Preemption hint for both RingBuffer and Batch Buffer can’t be enabled simultaneously.

<table>
<thead>
<tr>
<th><strong>DWord</strong></th>
<th><strong>Bit</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:2</td>
<td><strong>Preempted Hint Address</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: U30</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: GraphicsAddress[31:2]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This field contains the Head offset in to the Ring Buffer when Preemption Hint is set to Ring Buffer and Dword aligned Graphics Address in to the batch buffer when Preemption Hint is set to Batch Buffer.</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td><strong>Batch Buffer Preemption Hint</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: Enable</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Value</strong></th>
<th><strong>Name</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>Disabled</td>
<td>Preemption hint is disabled in batch buffer.</td>
</tr>
</tbody>
</table>
## VCS_PREEMPTION_HINT - VCS_PREEMPTION_HINT

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1h</td>
<td>Enabled</td>
<td>Preemption hint is enabled in ring buffer and preemption hint address</td>
</tr>
<tr>
<td></td>
<td></td>
<td>corresponds to the instruction in Batch Buffer.</td>
</tr>
<tr>
<td>0</td>
<td>Ring Preemption Hint</td>
<td>Format: Enable</td>
</tr>
<tr>
<td>0h</td>
<td>Disable</td>
<td>Preemption hint is disabled in ring buffer.</td>
</tr>
<tr>
<td>1h</td>
<td>Enabled</td>
<td>Preemption hint is enabled in ring buffer and preemption hint address</td>
</tr>
<tr>
<td></td>
<td></td>
<td>corresponds to the instruction in Ring Buffer.</td>
</tr>
</tbody>
</table>
## VCS_PREEMPTION_HINT_UDW

<table>
<thead>
<tr>
<th>VCS_PREEMPTION_HINT_UDW - VCS_PREEMPTION_HINT_UDW</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> VideoCS</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 124C8h</td>
</tr>
<tr>
<td><strong>Valid Projects:</strong> CHV, BSW</td>
</tr>
</tbody>
</table>

This register contains the 4GB aligned base address of gfx 4GB virtual address space within the host's 64-bit virtual address space of the Batch Buffer corresponding to MI_ARB_CHECK command called Preemption Hint Address.

### Programming Notes

**Programming Restriction:** This register should NEVER be programmed in functional mode, this should be used only in validation mode to achieve deterministic behavior of UH PTR being sampled by a given MI_ARB_CHK in command stream.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
<tr>
<td>15:0</td>
<td></td>
<td>Preempted Hint Address Upper DWORD</td>
</tr>
<tr>
<td></td>
<td>Format: GraphicsAddress[47:32]</td>
<td></td>
</tr>
</tbody>
</table>

This field contains the 4GB aligned base address of gfx 4GB virtual address space within the host's 64-bit virtual address space of the batch buffer when Preemption Hint is set to Batch Buffer. This field is not valid when Preemption Hint is set to Ring Buffer.
VCS Context ID Preemption Hint

**VCS_CTXID_PREEMPTION_HINT - VCS Context ID Preemption Hint**

<table>
<thead>
<tr>
<th>Register Space</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>VideoCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>124CCh</td>
</tr>
<tr>
<td>Valid Projects:</td>
<td>CHV, BSW</td>
</tr>
</tbody>
</table>

This register contains the Context ID of a context in execlist mode of operation. In execlist mode of operation VCS_PREEMPTION_HINT registers are looked at by Video Command Streamer on executing a context having Context ID that matches with the contents of this register. This register contents are valid and looked at only in execlist mode of operation.

**Programming Notes**

This register should NEVER be programmed in functional mode, this must be used only in validation mode to achieve deterministic preemption behavior in execlist mode of operation.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Context ID Preemption Hint</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format:</strong> U32</td>
</tr>
</tbody>
</table>

If 0 this field has no effect. If nonzero it indicates the only context ID that can be preempted when execlists are enabled. A preemption attempt when the context ID of the currently executing ring context does not match this field will be ignored.
VCS Context Sizes

**VCS_CXT_SIZE - VCS Context Sizes**

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
<th>Name</th>
<th>Project</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Reserved</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>VCS Context Size</strong></td>
<td>Ah</td>
<td>[Default]</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td><strong>VCR Context Size</strong></td>
<td>11h</td>
<td>[Default]</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td><strong>Execlist Context Size</strong></td>
<td>5h</td>
<td>[Default]</td>
<td>CHV, BSW</td>
</tr>
</tbody>
</table>

Register Space: MMIO: 0/2/0
Project: CHV, BSW
Source: VideoCS
Default Value: 0x000A1105 CHV, BSW
Access: Read/32 bit Write Only
Size (in bits): 32
Address: 121A8h
Valid Projects: CHV, BSW
VCS Context Timestamp Count

VCS_CTX_TIMESTAMP - VCS Context Timestamp Count

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>VideoCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>123A8h</td>
</tr>
<tr>
<td>Valid Projects:</td>
<td>CHV, BSW</td>
</tr>
</tbody>
</table>

This register provides a mechanism to obtain cumulative run time of a GPU context on HW. This register gets context save/restored on a context switch. SW must reset this register on very first submission of a context to HW, then afterwards gets context save/restored maintaining the cumulative run time of the corresponding context. This register (effectively) counts at a constant frequency by adjusting the increment amount according to the actual reference clock frequency. SW therefore does not need to know the reference clock frequency. This register gets reset on an engine reset.

This register is context save restore on a context switch.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Timestamp Value</strong></td>
</tr>
</tbody>
</table>

| Format | U32 |

This register increments for every 80 ns of time.
## VCS Counter for the bit stream decode engine

<table>
<thead>
<tr>
<th>VCS_CNTR - VCS Counter for the bit stream decode engine</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: VideoCS</td>
</tr>
<tr>
<td>Default Value: 0xFFFFFFFF</td>
</tr>
<tr>
<td>Access: R/W</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
</tbody>
</table>

Address: 12178h-1217Bh  
Valid Projects: CHV, BSW  

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Count Value</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: ffffffffh</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Writing a Zero value to this register starts the counting.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Writing a Value of FFFF FFFF to this counter stops the counter.</td>
</tr>
</tbody>
</table>
# VCS Error Identity Register

<table>
<thead>
<tr>
<th>VCS_EIR - VCS Error Identity Register</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: VideoCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Access: R/WC</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 120B0h</td>
</tr>
<tr>
<td>Valid Projects: CHV, BSW</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MBZ</td>
</tr>
<tr>
<td>15:0</td>
<td>Error Identity Bits</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Format: Array of Error condition bits see the table titled Hardware-Detected Error Bits</td>
<td></td>
</tr>
</tbody>
</table>

This register contains the persistent values of ESR error status bits that are unmasked via the EMR register. The logical OR of all (defined) bits in this register is reported in the Master Error bit of the Interrupt Status Register. In order to clear an error condition, software must first clear the error by writing a 1 to the appropriate bit(s) in this field. If required, software should then proceed to clear the Master Error bit of the IIR.

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>[Default]</td>
<td></td>
</tr>
<tr>
<td>1h</td>
<td>Error occurred</td>
<td>Error occurred</td>
</tr>
</tbody>
</table>

## Programming Notes

Writing a 1 to a set bit will cause that error condition to be cleared. However, the Page Table Error bit (Bit 4) cannot be cleared except by reset (i.e., it is a fatal error).
The EMR register is used by software to control which Error Status Register bits are "masked" or "unmasked". "Unmasked" bits will be reported in the EIR, thus setting the Master Error ISR bit and possibly triggering a CPU interrupt, and will persist in the EIR until cleared by software. "Masked" bits will not be reported in the EIR and therefore cannot generate Master Error conditions or CPU interrupts.

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td>31:16</td>
<td>Default Value: FFFFh</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: Must Be One</td>
</tr>
<tr>
<td>15:0</td>
<td></td>
<td><strong>Error Mask Bits</strong></td>
</tr>
<tr>
<td></td>
<td>0:15</td>
<td>Format: Array of error condition mask bits See the table titled Hardware-Detected Error Bits.</td>
</tr>
</tbody>
</table>

This register contains a bit mask that selects which error condition bits (from the ESR) are reported in the EIR.

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000h</td>
<td>Not Masked</td>
<td>Will be reported in the EIR</td>
</tr>
<tr>
<td>FFFFh</td>
<td>Masked [Default]</td>
<td>Will not be reported in the EIR</td>
</tr>
</tbody>
</table>
## VCS Error Status Register

<table>
<thead>
<tr>
<th>VCS_ESR - VCS Error Status Register</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: VideoCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Access: RO</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 120B8h</td>
</tr>
<tr>
<td>Valid Projects: CHV, BSW</td>
</tr>
</tbody>
</table>

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

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

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>[Default]</td>
<td></td>
</tr>
<tr>
<td>1h</td>
<td>Error Condition Detected</td>
<td>Error Condition detected</td>
</tr>
</tbody>
</table>
VCS Execute Condition Code Register

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>VideoCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W, RO</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Trusted Type:</td>
<td>1</td>
</tr>
<tr>
<td>Address:</td>
<td>12028h</td>
</tr>
<tr>
<td>Valid Projects:</td>
<td>CHV, BSW</td>
</tr>
</tbody>
</table>

This register contains user defined and hardware generated conditions that are used by MI_WAIT_FOR_EVENT commands. An MI_WAIT_FOR_EVENT instruction excludes the executing ring from arbitration if the selected event evaluates to a 1, while instruction is discarded if the condition evaluates to a 0. Once excluded, a ring is enabled into arbitration when the selected condition evaluates to a 0.

This register also contains control for the invalidation of indirect state pointers on context restore.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td><strong>Mask Bits</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: Mask[15:0]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>These bits serve as a write enable for bits 15:0. If this register is written with any of these bits clear the corresponding bit in the field 15:0 will not be modified. Reading these bits always returns 0s.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>15:5</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4:0</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
</tbody>
</table>
**VCS General Purpose Register**

<table>
<thead>
<tr>
<th><strong>VCS_GPR - VCS General Purpose Register</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space:</td>
</tr>
<tr>
<td>Project:</td>
</tr>
<tr>
<td>Source:</td>
</tr>
<tr>
<td>Default Value:</td>
</tr>
<tr>
<td>Access:</td>
</tr>
<tr>
<td>Size (in bits):</td>
</tr>
<tr>
<td>Address:</td>
</tr>
<tr>
<td>Valid Projects:</td>
</tr>
</tbody>
</table>

This is a general purpose register bank of sixteen 64-bit registers, used as temporary storage by the MI_MATH command to do ALU operations.

**Programming Notes**

Any operation that initiates a read to register 0x1266C will return the value of 0x1260c register. This does not include context save or MI_MATH command operation.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>63:0</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
</tbody>
</table>


VCS Hardware Status Mask Register

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>VideoCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0xFFFFFFFF</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Trusted Type:</td>
<td>1</td>
</tr>
<tr>
<td>Address:</td>
<td>12098h</td>
</tr>
<tr>
<td>Valid Projects:</td>
<td>CHV, BSW</td>
</tr>
</tbody>
</table>

Access: RO for Reserved Control bits

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

**Programming Notes**

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Hardware Status Mask Register</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FFFFFFFFh</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Array of Masks</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Refer to the table in the Interrupt Control Register section for bit definitions.</td>
</tr>
</tbody>
</table>
## VCS IDLE Max Count

<table>
<thead>
<tr>
<th>VCS_PWRCTX_MAXCNT - VCS IDLE Max Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: VideoCS</td>
</tr>
<tr>
<td>Default Value: 0x00000040 CHV, BSW</td>
</tr>
<tr>
<td>Access: R/W</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Trusted Type: 1</td>
</tr>
<tr>
<td>Address: 12054h</td>
</tr>
<tr>
<td>Valid Projects: CHV, BSW</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:20</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
<tr>
<td>19:0</td>
<td></td>
<td>Reserved</td>
</tr>
</tbody>
</table>
VCS Idle Switch Delay

<table>
<thead>
<tr>
<th>VCS_IDLEDLY - VCS Idle Switch Delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: VideoCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Access: R/W</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 1223Ch</td>
</tr>
<tr>
<td>Valid Projects: CHV, BSW</td>
</tr>
</tbody>
</table>

The IDLEDLY register contains an Idle Delay field which specifies the minimum number of microseconds allowed for command streamer to wait before a context is switched out leading to IDLE state in Execlist mode, i.e. following this context switch there is no active element available in HW to execute. A default value of 0, means that by default, there is no restriction to wait on a context switch leading to IDLE. This register has no significance when Execlists are not enabled.

### DWord Bit Description

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:21</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: All</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
<tr>
<td>20:0</td>
<td></td>
<td>IDLE Delay</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: All</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: U21</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Minimum number of micro-seconds allowed.</td>
</tr>
</tbody>
</table>
VCS Instruction Parser Mode Register

VCS_INSTPM - VCS Instruction Parser Mode Register

Register Space: MMIO: 0/2/0
Project: CHV, BSW
Source: VideoCS
Default Value: 0x00000000
Access: R/W
Size (in bits): 32
Address: 120C0h-120C3h
Valid Projects: CHV, BSW

The VCS_INSTPM register is used to control the operation of the VCS Instruction Parser. Certain classes of instructions can be disabled (ignored) - often useful for detecting performance bottlenecks. Also, "Synchronizing Flush" operations can be initiated - useful for ensuring the completion (vs. only parsing) of rendering instructions. DefaultValue=0000 0000h

Programming Notes

All reserved bits are implemented.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td><strong>Masks</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: Mask[15:0]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>These bits serve as write enables for bits 15:0. If this register is written with any of these bits clear the corresponding bit in the field 15:0 will not be modified. Reading these bits always returns 0s.</td>
</tr>
<tr>
<td>15:11</td>
<td><strong>Reserved</strong></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td><strong>Reserved</strong></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td><strong>Reserved</strong></td>
<td></td>
</tr>
<tr>
<td>8:7</td>
<td><strong>Reserved</strong></td>
<td></td>
</tr>
<tr>
<td>6:5</td>
<td><strong>Reserved</strong></td>
<td></td>
</tr>
<tr>
<td>4:0</td>
<td><strong>Reserved</strong></td>
<td></td>
</tr>
</tbody>
</table>

Format: MBZ
<table>
<thead>
<tr>
<th>VCS_INSTPM - VCS Instruction Parser Mode Register</th>
</tr>
</thead>
<tbody>
<tr>
<td>Format:</td>
</tr>
<tr>
<td>MBZ</td>
</tr>
</tbody>
</table>
VCS Interrupt Mask Register

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>VideoCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0xFFFFFFFF</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>120A8h</td>
</tr>
<tr>
<td>Valid Projects:</td>
<td>CHV, BSW</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>Interrupt Mask Bits</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: Array of interrupt mask bits Refer to the Interrupt Control Register section for bit definitions.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This field contains a bit mask which selects which interrupt bits (from the ISR) are reported in the IIR.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FFFF FFFFh</td>
<td>[Default]</td>
<td></td>
</tr>
<tr>
<td>0h</td>
<td>Not Masked</td>
<td>Will be reported in the IIR</td>
</tr>
<tr>
<td>1h</td>
<td>Masked</td>
<td>Will not be reported in the IIR</td>
</tr>
</tbody>
</table>
VCS Mode Register for Software Interface

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

### DWord | Bit | Description
---|---|---
0 | 31:16 | **Masks**
A 1 in a bit in this field allows the modification of the corresponding bit in Bits 15:0.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
</table>
| 15 | | **Suspend Flush**
| Mask: | MMIO(0x209c)#31 |
| Value | Name | Description |
| 0h | No Delay | HW will not delay flush, this bit will get cleared by MI_SUSPEND_FLUSH as well. |
| 1h | DelayFlush | Suspend flush is active |

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
</table>
| 14:12 | | **Reserved**
| Access: | R/W |

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
</table>
| 11 | | **Invalidate UHPTR enable**
If bit set H/W clears the valid bit of BCS_UHPTR (4134h, bit 0) when current active head pointer is equal to UHPTR.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
</table>
| 10 | | **Atomic Read Return for MI_COPY_MEM_MEM**
Project: | CHV, BSW |
Format: | U1 |
| Value | Name | Description |
| 0h | Disable | Hardware does a regular memory fence write to complete the write to the destination address before moving to the next instruction. |
| 1h | Enable | Hardware does Atomic Move with Read Return to complete the write to the destination address before moving to the next instruction. |
### VCS_MI_MODE - VCS Mode Register for Software Interface

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
<th>Value</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td><strong>Ring Idle (Read Only Status bit)</strong></td>
<td>RO</td>
<td>0</td>
<td>Parser not idle</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>Parser idle</td>
</tr>
</tbody>
</table>

*Writes to this bit are not allowed.*

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
<th>Value</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td><strong>Stop Ring</strong></td>
<td></td>
<td>0</td>
<td>Normal Operation</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>Parser is turned off</td>
</tr>
</tbody>
</table>

*Software must set this bit to force the Ring and Command Parser to Idle. Software must read a "1" in Ring Idle bit after setting this bit to ensure that the hardware is idle.*

*Software must clear this bit for Ring to resume normal operation.*

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Access</th>
<th>Value</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>7:0</td>
<td><strong>Reserved</strong></td>
<td>R/W</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Access:*}

- **RO**: Read Only
- **RW**: Read/Write
VCS Reported Timestamp Count

<table>
<thead>
<tr>
<th>VCS_TIMESTAMP - VCS Reported Timestamp Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: VideoCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000, 0x00000000</td>
</tr>
<tr>
<td>Access: RO. This register is not set by the context restore.</td>
</tr>
<tr>
<td>Size (in bits): 64</td>
</tr>
<tr>
<td>Address: 12358h</td>
</tr>
<tr>
<td>Valid Projects: CHV, BSW</td>
</tr>
</tbody>
</table>

This register provides an elapsed real-time value that can be used as a timestamp. This register is not reset by a graphics reset. It will maintain its value unless a full chipset reset is performed. Note: This timestamp register reflects the value of the PCU TSC. The PCU TSC counts 10ns increments; this timestamp reflects bits 38:3 of the TSC (i.e. 80ns granularity, rolling over every 1.5 hours).

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>63:36</td>
<td><strong>Reserved</strong></td>
<td>MBZ</td>
</tr>
<tr>
<td>35:0</td>
<td></td>
<td><strong>Timestamp Value</strong></td>
<td>U36</td>
</tr>
</tbody>
</table>

This register toggles every 80 ns. The upper 28 bits are zero.
VCS Reset Control Register

<table>
<thead>
<tr>
<th>VCS_RESET_CTRL - VCS Reset Control Register</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: VideoCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Access: R/W</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 120D0h</td>
</tr>
<tr>
<td>Valid Projects: CHV, BSW</td>
</tr>
</tbody>
</table>

This register is to be used to control soft reset.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td>Mask Bits</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: Mask[15:0]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Must be set to modify corresponding bit in Bits 15:0. (All implemented bits)</td>
</tr>
<tr>
<td>15:2</td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>Ready for Reset</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: U1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>When set indicates video codec engine is ready for reset. This bit gets cleared on engine reset or when Soft Reset In progress is cleared.</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td>Request Reset</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: U1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>When set indicates SW wishes to reset the video codec engine. On seeing this bit set Command Streamer will take appropriate action and set Ready For Reset status bit. This bit gets cleared on engine reset.</td>
</tr>
</tbody>
</table>
VCS Ring Buffer Next Context ID Register

<table>
<thead>
<tr>
<th>VCS_RNCID - VCS Ring Buffer Next Context ID Register</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: VideoCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000, 0x00000000</td>
</tr>
<tr>
<td>Access: R/W</td>
</tr>
<tr>
<td>Size (in bits): 64</td>
</tr>
<tr>
<td>Address: 12198h-1219Fh</td>
</tr>
<tr>
<td>Valid Projects: CHV, BSW</td>
</tr>
</tbody>
</table>

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

### Programming Notes

The current context (RCCID) register can be updated indirectly from this register on a context switch event. Note that this can only be triggered when arbitration is enabled or if the current context runs dry (head pointer becomes equal to tail pointer).

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>63:0</td>
<td><strong>Context ID</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>See Context Descriptor for VCS.</td>
</tr>
</tbody>
</table>
## VCS Semaphore Polling Interval on Wait

**VCS_SEMA_WAIT_POLL - VCS Semaphore Polling Interval on Wait**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>VideoCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>1224Ch</td>
</tr>
<tr>
<td>Valid Projects:</td>
<td>CHV, BSW</td>
</tr>
</tbody>
</table>

The SEMA_WAIT_POLL register contains Poll Interval field which specifies the minimum number of microseconds allowed for command streamer to wait before re-fetching the data from the address mentioned in the MI_SEMAPHORE_WAIT command on WAIT Mode set to POLL until the condition is satisfied while the context is not switched out. When value of 0 is written the poll interval will be equal to the memory latency of the read completion.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:21</td>
<td><strong>Reserved</strong></td>
<td>MBZ</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Poll Interval</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>20:0</td>
<td>Minimum number of micro-seconds allowed</td>
<td></td>
</tr>
</tbody>
</table>
VCS Threshold for the counter of bit stream decode engine

<table>
<thead>
<tr>
<th>VCS_THRSH - VCS Threshold for the counter of bit stream decode engine</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: VideoCS</td>
</tr>
<tr>
<td>Default Value: 0x00150000</td>
</tr>
<tr>
<td>Access: R/W</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 1217Ch-1217Fh</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Threshold Value</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00150000h</td>
</tr>
</tbody>
</table>

The value in this register reflects the number of clocks the bit stream decode engine is expected to run. If the value is exceeded the counter is reset and an interrupt may be enabled in the device.
VCW Clock Count

### VCW_CLOCK_CNT - VCW Clock Count

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>VideoEnhancementCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>RO</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Trusted Type:</td>
<td>1</td>
</tr>
<tr>
<td>Address:</td>
<td>08820h</td>
</tr>
<tr>
<td>ShortName:</td>
<td>VCW0_CLOCK_CNT</td>
</tr>
<tr>
<td>Address:</td>
<td>08920h</td>
</tr>
<tr>
<td>ShortName:</td>
<td>VCW1_CLOCK_CNT</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
<tr>
<td>23:0</td>
<td>Max clock count</td>
<td>Default Value: 0h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Maximum number of clocks taken by VCW to process a column</td>
</tr>
</tbody>
</table>
# VCW Internal Latency

## VCW_INTERNAL_LAT - VCW Internal Latency

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Format:</th>
<th>Default Value:</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td>Reserved</td>
<td>MBZ</td>
<td>0h</td>
</tr>
<tr>
<td></td>
<td>23:0</td>
<td>VCW internal data latency count</td>
<td></td>
<td>0h</td>
</tr>
</tbody>
</table>

- **Register Space:** MMIO: 0/2/0
- **Project:** CHV, BSW
- **Source:** VideoEnhancementCS
- **Default Value:** 0x00000000
- **Access:** RO
- **Size (in bits):** 32
- **Trusted Type:** 1
- **Address:** 08824h
- **ShortName:** VCW0_INTERNAL_LAT
- **Address:** 08924h
- **ShortName:** VCW1_INTERNAL_LAT
# VCW Min Max Latency

<table>
<thead>
<tr>
<th>VCW_MINMAX_LAT - VCW Min Max Latency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: VideoEnhancementCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Access: RO</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Trusted Type: 1</td>
</tr>
<tr>
<td>Address: 08828h</td>
</tr>
<tr>
<td>ShortName: VCW0_MINMAX_LAT</td>
</tr>
<tr>
<td>Address: 08928h</td>
</tr>
<tr>
<td>ShortName: VCW1_MINMAX_LAT</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td><strong>Current request count</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0h</td>
</tr>
<tr>
<td>0</td>
<td>15:8</td>
<td><strong>Max latency</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Maximum number of clocks taken for tag 200h</td>
</tr>
<tr>
<td>0</td>
<td>7:0</td>
<td><strong>Min latency</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Minimum number of clocks taken for tag 200h</td>
</tr>
</tbody>
</table>
## VCW Total Latency

<table>
<thead>
<tr>
<th>VCW_TOTAL_LAT - VCW Total Latency</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> VideoEnhancementCS</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Trusted Type:</strong> 1</td>
</tr>
<tr>
<td><strong>Address:</strong> 0882Ch</td>
</tr>
<tr>
<td><strong>ShortName:</strong> VCW0_TOTAL_LAT</td>
</tr>
<tr>
<td><strong>Address:</strong> 0892Ch</td>
</tr>
<tr>
<td><strong>ShortName:</strong> VCW1_TOTAL_LAT</td>
</tr>
<tr>
<td><strong>DWORD</strong></td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
## VCW XY position

<table>
<thead>
<tr>
<th>VCW_XY_POS - VCW XY position</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: VideoEnhancementCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Access: RO</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Trusted Type: 1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Address</th>
<th>08830h</th>
</tr>
</thead>
<tbody>
<tr>
<td>ShortName</td>
<td>VCW0_XY_POS</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Address</th>
<th>08930h</th>
</tr>
</thead>
<tbody>
<tr>
<td>ShortName</td>
<td>VCW1_XY_POS</td>
</tr>
</tbody>
</table>

### DWord | Bit | Description |
|---------|-----|-------------|
| 0       | 31:16 | **Current Y value**  
|         |      | Default Value: 0h  
|         | 15:0 | **Current X value**  
|         |      | Default Value: 0h  

- Current Y position of VCW walker
- Current X position of VCW walker
## VEBOX TLB Control Register

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:1</td>
<td><strong>Reserved</strong></td>
<td>0000000000000000000000000000000b</td>
<td>RO</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td><strong>Invalidate TLBs on the corresponding Engine</strong></td>
<td>0b</td>
<td>R/W</td>
</tr>
</tbody>
</table>

SW writes 1 to invalidate the TLBs for the associated engine and HW clears the bit when invalidation is complete. To ensure proper invalidation of the TLBs, SW has to ensure the corresponding engine's HW pipeline is flushed and cleared from all its memory accesses. Otherwise HW cannot guarantee the proper invalidation for TLBs. This bit is self clear.
# VEBX Context Element Descriptor (High Part)

<table>
<thead>
<tr>
<th>VEBX_CTX_EDR_H - VEBX Context Element Descriptor (High Part)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 044C4h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>VEBX Context Element Descriptor (High Part)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>
# VEBX Context Element Descriptor (Low Part)

<table>
<thead>
<tr>
<th>VEBX_CTX_EDR_L - VEBX Context Element Descriptor (Low Part)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000009</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 044C0h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>VEBX Context Element Descriptor (Low Part)</td>
<td>00000009h</td>
<td>R/W</td>
</tr>
</tbody>
</table>
## VEBX Context Element Descriptor (Low Part)

<table>
<thead>
<tr>
<th>VEBX_CTX_EDR_L - VEBX Context Element Descriptor (Low Part)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space:</td>
</tr>
<tr>
<td>Project:</td>
</tr>
<tr>
<td>Source:</td>
</tr>
<tr>
<td>Default Value:</td>
</tr>
<tr>
<td>Size (in bits):</td>
</tr>
<tr>
<td>Address:</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>VEBX Context Element Descriptor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000009h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>


# VEBX LRA 0

**Register Space:** MMIO: 0/2/0  
**Project:** CHV, BSW  
**Source:** PRM  
**Default Value:** 0x2F201F00  
**Size (in bits):** 32  
**Address:** 04A80h

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:30</td>
<td><strong>Reserved</strong></td>
<td>00b</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td>29:24</td>
<td><strong>VEBX LRA1 Max</strong></td>
<td>101111b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Maximum value of programmable LRA1.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>23:22</td>
<td><strong>Reserved</strong></td>
<td>00b</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td>21:16</td>
<td><strong>VEBX LRA1 Min</strong></td>
<td>100000b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Minimum value of programmable LRA1.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>15:14</td>
<td><strong>Reserved</strong></td>
<td></td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td>13:8</td>
<td><strong>VEBX LRA0 Max</strong></td>
<td>011111b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Maximum value of programmable LRA0.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>7:6</td>
<td><strong>Reserved</strong></td>
<td>00b</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td>5:0</td>
<td><strong>VEBX LRA0 Min</strong></td>
<td>000000b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Minimum value of programmable LRA0.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DWord</td>
<td>Bit</td>
<td>Description</td>
<td>Default Value:</td>
<td>Access:</td>
</tr>
<tr>
<td>-------</td>
<td>------</td>
<td>------------------------------</td>
<td>--------------------</td>
<td>---------</td>
</tr>
<tr>
<td>0</td>
<td>31:30</td>
<td>Reserved</td>
<td>00b</td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>29:28</td>
<td>VECs</td>
<td></td>
<td>00b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Which LRA should VECS use.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>27:26</td>
<td>VFW</td>
<td></td>
<td>01b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Which LRA should VFW use.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25:24</td>
<td>VEO</td>
<td></td>
<td>10b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Which LRA should VEO use.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23:14</td>
<td>Reserved</td>
<td></td>
<td>0000000000b</td>
<td>RO</td>
</tr>
<tr>
<td>13:8</td>
<td>VEBXLRA2 Max</td>
<td></td>
<td>111111b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Minimum value of programmable LRA2.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7:6</td>
<td>Reserved</td>
<td></td>
<td>00b</td>
<td>RO</td>
</tr>
</tbody>
</table>
### VEBX_LRA_1 - VEBX LRA 1

<table>
<thead>
<tr>
<th>5:0</th>
<th>VEBXLRA2 Min</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Default Value: 110000b</td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

Minimum value of programmable LRA2.
# VEBX PDP0/PML4/PASID Descriptor (High Part)

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>044CCh</td>
</tr>
</tbody>
</table>

## VEBX_CTX_PDP0_H - VEBX PDP0/PML4/PASID Descriptor (High Part)

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>VEBX PDP0/PML4/PASID Descriptor (High Part)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>


### VEBX PDP0/PML4/PASID Descriptor (Low Part)

<table>
<thead>
<tr>
<th>VEBX_CTX_PDP0_L - VEBX PDP0/PML4/PASID Descriptor (Low Part)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 044C8h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>VEBX PDP0/PML4/PASID Descriptor (Low Part)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

## VEBX PDP1 Descriptor Register (High Part)

<table>
<thead>
<tr>
<th>VEBX_CTX_PDP1_H - VEBX PDP1 Descriptor Register (High Part)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 044D4h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>VEBX PDP1 Descriptor Register (High Part)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

[Image of the table]
### VEBX PDP1 Descriptor Register (Low Part)

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>VEBX PDP1 Descriptor Register (Low Part)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

Register Space: MMIO: 0/2/0  
Project: CHV, BSW  
Source: PRM  
Default Value: 0x00000000  
Size (in bits): 32  
Address: 044D0h
### VEBX PDP2 Descriptor Register (High Part)

<table>
<thead>
<tr>
<th>VEBX_CTX_PDP2_H - VEBX PDP2 Descriptor Register (High Part)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong></td>
</tr>
<tr>
<td>CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong></td>
</tr>
<tr>
<td>PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong></td>
</tr>
<tr>
<td>0x00000000h</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong></td>
</tr>
<tr>
<td>32</td>
</tr>
</tbody>
</table>

**Address:** 044DCh

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>VEBX PDP2 Descriptor Register (High Part)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
</tbody>
</table>
### VEBX PDP2 Descriptor Register (Low Part)

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space</td>
<td>MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits)</td>
<td>32</td>
</tr>
<tr>
<td>Address</td>
<td>044D8h</td>
</tr>
</tbody>
</table>

#### Description

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>VEBX PDP2 Descriptor Register (Low Part)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>
VEBX PDP3 Descriptor Register (High Part)

<table>
<thead>
<tr>
<th>VEBX_CTX_PDP3_H - VEBX PDP3 Descriptor Register (High Part)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x000000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 044E4h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>VEBX PDP3 Descriptor Register (High Part)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>
## VEBX PDP3 Descriptor Register (Low Part)

<table>
<thead>
<tr>
<th>VEBX_CTX_PDP3_L - VEBX PDP3 Descriptor Register (Low Part)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> PRM</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 044E0h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>VEBX PDP3 Descriptor Register (Low Part)</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Access:</strong> R/W</td>
</tr>
</tbody>
</table>
**VECS_PREEMPTION_HINT**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>VideoEnhancementCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>1A4BCh</td>
</tr>
</tbody>
</table>

This register contains the Head pointer offset in to the Ring Buffer or the Dword aligned Graphics address in to the Batch Buffer corresponding to either MI_ARB_CHECK called Preemption Hint Address. When Preemption Hint Address is enabled, VECS will honor UHPTR only on parsing MI_ARB_CHK at Preemption Hint Address.

This register contains the Head pointer offset into the Ring Buffer or the Dword aligned Graphics address into the Batch Buffer corresponding to the below listed commands in executable mode of operation:

- MI_ARB_CHECK
- MI_WAIT_FOR_EVENT
- MI_SEMAPHORE_WAIT

**Programming Notes**

**Programming Restriction:** This register should NEVER be programmed in functional mode, this should be used only in validation mode to achieve deterministic behavior of UHPTR being sampled by a given MI_ARB_CHECK in command stream. Programmer has to ensure that VECS Preemption Hint register gets programmed before UHPTR is programmed and well before VECS crosses the corresponding execution point. Preemption hint for both RingBuffer and Batch Buffer can’t be enabled simultaneously.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:2</td>
<td><strong>Preempted Hint Address</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: U30</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: GraphicsAddress[31:2]</td>
</tr>
</tbody>
</table>

This field contains the Head offset in to the Ring Buffer when Preemption Hint is set to Ring Buffer and Dword aligned Graphics Address in to the batch buffer when Preemption Hint is set to Batch Buffer.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0h</td>
<td>Disabled</td>
<td>Preemption hint is disabled in batch buffer.</td>
</tr>
<tr>
<td></td>
<td>1h</td>
<td>Enabled</td>
<td>Preemption hint is enabled in ring buffer and preemption hint address corresponds to the instruction in Batch Buffer.</td>
</tr>
</tbody>
</table>
### VECS_PREEMPTION_HINT - VECS_PREEMPTION_HINT

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>Disable</td>
<td>Preemption hint is disabled in ring buffer.</td>
</tr>
<tr>
<td>1h</td>
<td>Enabled</td>
<td>Preemption hint is enabled in ring buffer and preemption hint address</td>
</tr>
<tr>
<td></td>
<td></td>
<td>corresponds to the instruction in Ring Buffer.</td>
</tr>
</tbody>
</table>

**Ring Preemption Hint**

Format: Enable
VECS Context ID Preemption Hint

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>VideoEnhancementCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>1A4CCh</td>
</tr>
</tbody>
</table>

This register contains the Context ID of a context in Execlist mode of operation. In execlist mode of operation, VECS_PREEMPTION_HINT registers are looked at by Video Enhancement Command Streamer on executing a context having Context ID that matches with the contents of this register. This register contents are valid and looked at only in Execlist mode of operation.

### Programming Notes

This register should NEVER be programmed in functional mode, this must be used only in validation mode to achieve deterministic preemption behavior in execlist mode of operation.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Context ID Preemption Hint</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: U32</td>
</tr>
</tbody>
</table>

If 0 this field has no effect. If nonzero it indicates the only context ID that can be preempted when execlists are enabled. A preemption attempt when the context ID of the currently executing ring context does not match this field will be ignored.
# VECS Context Timestamp Count

<table>
<thead>
<tr>
<th><strong>Register Space:</strong></th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Project:</strong></td>
<td>CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong></td>
<td>VideoEnhancementCS</td>
</tr>
<tr>
<td><strong>Default Value:</strong></td>
<td>0x00000000</td>
</tr>
<tr>
<td><strong>Access:</strong></td>
<td>R/W</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong></td>
<td>32</td>
</tr>
<tr>
<td><strong>Address:</strong></td>
<td>1A3A8h</td>
</tr>
</tbody>
</table>

This register provides a mechanism to obtain cumulative run time of a GPU context on HW. This register gets context save/restored on a context switch. SW must reset this register on very first submission of a context to HW, then afterwards gets context save/restored maintaining the cumulative run time of the corresponding context. Note that the value of this register can be obtained in a 3D pipeline-synchronous fashion without a pipeline flush by using the PIPE_CONTROL command. See 3D Geometry Pipeline in the "3D and Media" volume. This register (effectively) counts at a constant frequency by adjusting the increment amount according to the actual reference clock frequency. SW therefore does not need to know the reference clock frequency. This register gets reset on an engine reset.

This register is context save restore on a context switch.

<table>
<thead>
<tr>
<th><strong>DWord</strong></th>
<th><strong>Bit</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Timestamp Value</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: U32</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This register increments for every 80 ns of time.</td>
</tr>
</tbody>
</table>
VECS Counter for the Video Enhancement Engine

<table>
<thead>
<tr>
<th>VECS_CNTR - VECS Counter for the Video Enhancement Engine</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: VideoEnhancementCS</td>
</tr>
<tr>
<td>Default Value: 0xFFFFFFFF</td>
</tr>
<tr>
<td>Access: R/W</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 1A178h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Count Value</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: ffffffffh</td>
</tr>
</tbody>
</table>

Writing a Zero value to this register starts the counting. Writing a Value of FFFF FFFF to this counter stops the counter.
## VECS Error Identity Register

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: All</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
<tr>
<td>15:0</td>
<td><strong>Error Identity Bits</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Project: All</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Format: Array of Error condition bits See Table 15. Hardware-Detected Error Bits</td>
<td></td>
</tr>
</tbody>
</table>

This register contains the persistent values of ESR error status bits that are unmasked via the EMR register. (See Error! Reference source not found.). The logical OR of all (defined) bits in this register is reported in the Master Error bit of the Interrupt Status Register. In order to clear an error condition, software must first clear the error by writing a ‘1’ to the appropriate bit(s) in this field. If required, software should then proceed to clear the Master Error bit of the IIR.

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
<th>Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td><strong>[Default]</strong></td>
<td>Error occurred</td>
<td>All</td>
</tr>
<tr>
<td>1h</td>
<td>Error occurred</td>
<td>Error occurred</td>
<td>All</td>
</tr>
</tbody>
</table>

**Programming Notes**

Writing a ‘1’ to a set bit will cause that error condition to be cleared. However, the Page Table Error bit (Bit 4) can not be cleared except by reset (i.e., it is a fatal error).
# VECS Error Mask Register

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>VideoEnhancementCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0xFFFFFFFF CHV, BSW</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>1A0B4h</td>
</tr>
</tbody>
</table>

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

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value:  FFFFh</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: Must Be One</td>
</tr>
<tr>
<td>15:0</td>
<td>Error Mask Bits</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Project: All</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Format: Array of error condition mask bits See Table 1 5. Hardware-Detected Error Bits</td>
<td></td>
</tr>
</tbody>
</table>

This register contains a bit mask that selects which error condition bits (from the ESR) are reported in the EIR.

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000h</td>
<td>Not Masked</td>
<td>Will be reported in the EIR</td>
</tr>
<tr>
<td>FFFFh</td>
<td>Masked [Default]</td>
<td>Will not be reported in the EIR</td>
</tr>
</tbody>
</table>
**VECS Error Status Register**

<table>
<thead>
<tr>
<th>Register Space</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source</td>
<td>VideoEnhancementCS</td>
</tr>
<tr>
<td>Default Value</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access</td>
<td>RO</td>
</tr>
<tr>
<td>Size (in bits)</td>
<td>32</td>
</tr>
<tr>
<td>Address</td>
<td>1A0B8h</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: All</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
<tr>
<td>15:0</td>
<td></td>
<td><strong>Error Status Bits</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: All</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: Array of error condition bits See Table 1 5. Hardware-Detected Error Bits</td>
</tr>
</tbody>
</table>

This register contains the non-persistent values of all hardware-detected error condition bits.

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
<th>Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>[Default]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1h</td>
<td>Error Condition Detected</td>
<td>Error Condition detected</td>
<td>All</td>
</tr>
</tbody>
</table>
VECS General Purpose Register

<table>
<thead>
<tr>
<th>VECS_GPR - VECS General Purpose Register</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: VideoEnhancementCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000, 0x00000000</td>
</tr>
<tr>
<td>Access: R/W</td>
</tr>
<tr>
<td>Size (in bits): 64</td>
</tr>
<tr>
<td>Address: 1A600h-1A67Fh</td>
</tr>
</tbody>
</table>

This is a general purpose register bank of sixteen 64-bit registers, used as temporary storage by the MI_MATH command to do ALU operations.

<table>
<thead>
<tr>
<th>Programming Notes</th>
<th>Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>Any operation that initiates a read to register 0x1A66C will return the value of 0x1A60c register. This does not include context save or MI_MATH command operation.</td>
<td>CHV, BSW</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>63:0</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

Format: MBZ
### VECS Hardware Status Mask Register

<table>
<thead>
<tr>
<th><strong>VECS_HWSTAM - VECS Hardware Status Mask Register</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> VideoEnhancementCS</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0xFFFFFFFF</td>
</tr>
<tr>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Trusted Type:</strong> 1</td>
</tr>
<tr>
<td><strong>Address:</strong> 1A098h</td>
</tr>
</tbody>
</table>

**Access: RO for Reserved Control bits**

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

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

<table>
<thead>
<tr>
<th><strong>DWord</strong></th>
<th><strong>Bit</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Hardware Status Mask Register</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0xFFFFFFFFh</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: All</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: Array of Masks</td>
</tr>
<tr>
<td></td>
<td></td>
<td>refer to Table 4-4 in Interrupt Control Register section for bit definitions</td>
</tr>
</tbody>
</table>
## VECS IDLE Max Count

<table>
<thead>
<tr>
<th>VECS_PWRCTX_MAXCNT - VECS IDLE Max Count</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> VideoEnhancementCS</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000040</td>
</tr>
<tr>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Trusted Type:</strong> 1</td>
</tr>
</tbody>
</table>

**Address:** 1A054h

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:20</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td>Project:</td>
<td>All</td>
</tr>
<tr>
<td></td>
<td>Format:</td>
<td>MBZ</td>
</tr>
<tr>
<td>19:0</td>
<td><strong>Reserved</strong></td>
<td></td>
</tr>
</tbody>
</table>

---

**Register Space:** MMIO: 0/2/0  
**Project:** CHV, BSW  
**Source:** VideoEnhancementCS  
**Default Value:** 0x00000040  
**Access:** R/W  
**Size (in bits):** 32  
**Trusted Type:** 1  
**Address:** 1A054h  

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:20</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td>Project:</td>
<td>All</td>
</tr>
<tr>
<td></td>
<td>Format:</td>
<td>MBZ</td>
</tr>
<tr>
<td>19:0</td>
<td><strong>Reserved</strong></td>
<td></td>
</tr>
</tbody>
</table>
VECS Instruction Parser Mode Register

### VECS_INSTPM - VECS Instruction Parser Mode Register

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>VideoEnhancementCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>1A0C0h-1A0C3h</td>
</tr>
</tbody>
</table>

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

### Programming Notes

All reserved bits are implemented

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td>Masks</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: Mask[15:0]</td>
</tr>
</tbody>
</table>

These bits serve as write enables for bits 15:0. If this register is written with any of these bits clear the corresponding bit in the field 15:0 will not be modified. Reading these bits always returns 0s.

| 15:11 | Reserved |
|       | Project: All |
|       | Format: MBZ |

| 10    | Reserved |
|       | Project: CHV, BSW |
|       | Format: MBZ |

| 9     | Reserved |
|       | Project: CHV, BSW |
|       | Format: MBZ |

| 8:7   | Reserved |
|       | Format: MBZ |

| 6:5   | Reserved |
|       | Project: CHV, BSW |
|       | Format: MBZ |

| 4:0   | Reserved |
## VECS Interrupt Mask Register

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>VideoEnhancementCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0xFFFFFFFF</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>1A0A8h</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>Interrupt Mask Bits</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: All</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: Array of interrupt mask bits Refer to Table 4-4 in Interrupt Control Register section for bit definitions</td>
</tr>
</tbody>
</table>

This field contains a bit mask which selects which interrupt bits (from the ISR) are reported in the IIR.

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
<th>Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>FFFF FFFh</td>
<td>[Default]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0h</td>
<td>Not Masked</td>
<td>Will be reported in the IIR</td>
<td>All</td>
</tr>
<tr>
<td>1h</td>
<td>Masked</td>
<td>Will not be reported in the IIR</td>
<td>All</td>
</tr>
</tbody>
</table>
### VECS Mode Register for Software Interface

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>VideoEnhancementCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000200 CHV, BSW</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>1A09Ch-1A09Fh</td>
</tr>
</tbody>
</table>

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

#### DWord 0

**Bit 31:16**
**Masks**
A "1" in a bit in this field allows the modification of the corresponding bit in Bits 15:0.

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>No Delay</td>
<td>HW will not delay flush, this bit will get cleared by MI_SUSPEND_FLUSH as</td>
</tr>
<tr>
<td></td>
<td></td>
<td>well</td>
</tr>
<tr>
<td>1h</td>
<td>Delay</td>
<td>Suspend flush is active</td>
</tr>
</tbody>
</table>

#### DWord 15

**Bit 15**
**Suspend Flush**

<table>
<thead>
<tr>
<th>Project:</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mask:</td>
<td>MMIO(0x209c)#31</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>No Delay</td>
<td>HW will not delay flush, this bit will get cleared by MI_SUSPEND_FLUSH as</td>
</tr>
<tr>
<td></td>
<td></td>
<td>well</td>
</tr>
<tr>
<td>1h</td>
<td>Delay</td>
<td>Suspend flush is active</td>
</tr>
</tbody>
</table>

#### DWord 14:12

**Reserved**

| Format: | MBZ       |

#### DWord 11

**Invalidate UHPTR Enable**

If bit set H/W clears the valid bit of BCS_UHPTR (4134h, bit 0) when current active head pointer is equal to UHPTR.

#### DWord 10

**Atomic Read Return for MI_COPY_MEM_MEM**

<table>
<thead>
<tr>
<th>Project:</th>
<th>CHV, BSW</th>
</tr>
</thead>
<tbody>
<tr>
<td>Format:</td>
<td>U1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>Disable</td>
<td>Hardware does a regular memory fence write to complete the write to the</td>
</tr>
<tr>
<td></td>
<td>[Default]</td>
<td>destination address before moving to the next instruction.</td>
</tr>
<tr>
<td>1h</td>
<td>Enable</td>
<td>Hardware does Atomic Move with Read Return to complete the write to the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>destination address before moving to the next instruction.</td>
</tr>
</tbody>
</table>
### VECS_MI_MODE - VECS Mode Register for Software Interface

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>9</strong></td>
<td><strong>Ring Idle (Read Only Status bit)</strong></td>
<td><em>W</em>rites to this bit are not allowed.*</td>
</tr>
<tr>
<td>Value</td>
<td>Name</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>Parser not idle</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Parser idle [Default]</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>8</strong></th>
<th><strong>Stop Ring</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Normal Operation.</td>
</tr>
<tr>
<td>1</td>
<td>Parser is turned off.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th><strong>7:0</strong></th>
<th><strong>Reserved</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Format:</td>
<td>MBZ</td>
</tr>
</tbody>
</table>
## VECS PREEMPTION HINT UDW

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>VideoEnhancementCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>1A4C8h</td>
</tr>
</tbody>
</table>

This register contains the 4GB aligned base address of gfx 4GB virtual address space within the host’s 64-bit virtual address space of the Batch Buffer corresponding to MI_ARB_CHECK command called Preemption Hint Address.

### Programming Notes

**Programming Restriction:** This register should NEVER be programmed in functional mode, this should be used only in validation mode to achieve deterministic behavior of UHPTTR being sampled by a given MI_ARB_CHK in command stream.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td>Reserved</td>
<td>MBZ</td>
<td></td>
</tr>
<tr>
<td>15:0</td>
<td></td>
<td>Preempted Hint Address Upper DWORD</td>
<td>GraphicsAddress[47:32]</td>
<td></td>
</tr>
</tbody>
</table>

This field contains the 4GB aligned base address of gfx 4GB virtual address space within the host’s 64-bit virtual address space of the batch buffer when Preemption Hint is set to Batch Buffer. This field is not valid when Preemption Hint is set to Ring Buffer.
VECS Reported Timestamp Count

<table>
<thead>
<tr>
<th>VECS_TIMESTAMP - VECS Reported Timestamp Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: VideoEnhancementCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000, 0x00000000</td>
</tr>
<tr>
<td>Access: RO. This register is not set by the context restore.</td>
</tr>
<tr>
<td>Size (in bits): 64</td>
</tr>
<tr>
<td>Address: 1A358h</td>
</tr>
</tbody>
</table>

This register provides an elapsed real-time value that can be used as a timestamp. This register is not reset by a graphics reset. It will maintain its value unless a full chipset reset is performed. Note: This timestamp register reflects the value of the PCU TSC. The PCU TSC counts 10ns increments; this timestamp reflects bits 38:3 of the TSC (i.e. 80ns granularity, rolling over every 1.5 hours).

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>63:36</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: All</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
<tr>
<td>35:0</td>
<td>TimeStampValue</td>
<td>Project: All</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: U36</td>
</tr>
</tbody>
</table>

This register toggles every 80 ns. The upper 28 bits are zero.
VECS Reset Control Register

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>VideoEnhancementCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>1A0D0h</td>
</tr>
</tbody>
</table>

This register is to be used to control soft reset.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td>Mask Bits</td>
</tr>
<tr>
<td></td>
<td>Format: Mask[15:0]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Must be set to modify corresponding bit in Bits 15:0. (All implemented bits)</td>
<td></td>
</tr>
<tr>
<td>15:2</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Format: MBZ</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Ready for Reset</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Format: U1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>When set indicates video enhancement engine is ready for reset. This bit gets cleared on engine reset or when Soft Reset In progress is cleared.</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>Request Reset</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Format: U1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>When set indicates SW wishes to reset the video enhancement engine. On seeing this bit set Command Streamer will take appropriate action and set Ready For Reset status bit. This bit gets cleared on engine reset.</td>
<td></td>
</tr>
</tbody>
</table>
VECS Semaphore Polling Interval on Wait

<table>
<thead>
<tr>
<th><strong>VECS_SEMA_WAIT_POLL - VECS Semaphore Polling Interval on Wait</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> VideoEnhancementCS</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Address:</strong> 1A24Ch</td>
</tr>
</tbody>
</table>

The SEMA_WAIT_POLL register contains Poll Interval field which specifies the minimum number of microseconds allowed for command streamer to wait before re-fetching the data from the address mentioned in the MI_SEMAPHORE_WAIT command on WAIT Mode set to POLL until the condition is satisfied while the context is not switched out. When value of 0 is written the poll interval will be equal to the memory latency of the read completion.

<table>
<thead>
<tr>
<th><strong>DWord</strong></th>
<th><strong>Bit</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:21</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format:</strong> MBZ</td>
</tr>
<tr>
<td></td>
<td>20:0</td>
<td><strong>Poll Interval</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Minimum number of micro-seconds allowed</td>
</tr>
</tbody>
</table>

VECS Threshold for the Counter of Video Enhancement Engine

**VECS_CTR_THRSH - VECS Threshold for the Counter of Video Enhancement Engine**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>VideoEnhancementCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00150000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>1A17Ch</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Threshold Value</strong></td>
</tr>
</tbody>
</table>

Default Value: 00150000h

The value in this register reflects the number of clocks the bit stream decode engine is expected to run. If the value is exceeded the counter is reset and an interrupt may be enabled in the device.
## VEO Current Pipe 0 XY Register

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:30</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td></td>
<td>29:16</td>
<td>Current Input Pipe 0 X</td>
<td>0h</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td></td>
<td>14:0</td>
<td>Current Input Pipe 0 Y</td>
<td>0h</td>
</tr>
</tbody>
</table>
# VEO DN Pipe 0 XY Register

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>VideoEnhancementCS</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>RO</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Trusted Type:</td>
<td>1</td>
</tr>
</tbody>
</table>

Address: 0884Ch

Address: 0894Ch

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:30</td>
<td>Reserved</td>
</tr>
<tr>
<td>29:16</td>
<td></td>
<td>DN Pipe 0 X</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>dn_input_x[13:0]</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td>14:0</td>
<td></td>
<td>DN Pipe 0 Y</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>dn_input_y[14:0]</td>
</tr>
</tbody>
</table>
# VEO DN Pipe 1 XY Register

<table>
<thead>
<tr>
<th>VEO_DN1_XY - VEO DN Pipe 1 XY Register</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: VideoEnhancementCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Access: RO</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Trusted Type: 1</td>
</tr>
<tr>
<td>Address: 08850h</td>
</tr>
<tr>
<td>Address: 08950h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>31:30</td>
<td><strong>Reserved</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>29:16</td>
<td><strong>DN Pipe 1 X</strong></td>
<td><strong>Default Value:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0h</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td><strong>Reserved</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>14:0</td>
<td><strong>DN Pipe 1 Y</strong></td>
<td><strong>Default Value:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0h</td>
</tr>
</tbody>
</table>
### VEO DV Count Register

<table>
<thead>
<tr>
<th>VEO_DV_COUNT - VEO DV Count Register</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> VideoEnhancementCS</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Trusted Type:</strong> 1</td>
</tr>
<tr>
<td><strong>Address:</strong> 08844h</td>
</tr>
<tr>
<td><strong>Address:</strong> 08944h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:24</td>
<td><strong>Pipe1 MotinoDV/Hold Maxcount</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0h</td>
</tr>
<tr>
<td></td>
<td>23:16</td>
<td><strong>Pipe1 Pixel History DV/Hold Maxcount</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0h</td>
</tr>
<tr>
<td></td>
<td>15:8</td>
<td><strong>Pipe0 Motion History DV/Hold Maxcount</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0h</td>
</tr>
<tr>
<td></td>
<td>7:0</td>
<td><strong>Pipe0 Pixel History DV/Hold Maxcount</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0h</td>
</tr>
</tbody>
</table>
## VEO DV Hold Register

<table>
<thead>
<tr>
<th>VEO_DVHOLD - VEO DV Hold Register</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> VideoEnhancementCS</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000</td>
</tr>
<tr>
<td><strong>Access:</strong> RO</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 32</td>
</tr>
<tr>
<td><strong>Trusted Type:</strong> 1</td>
</tr>
<tr>
<td><strong>Address:</strong> 0885Ch</td>
</tr>
<tr>
<td><strong>Address:</strong> 0895Ch</td>
</tr>
<tr>
<td><strong>Datavalid/Hold signals for VEO interface</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31</td>
<td>vdn_p0_veo_pixel_dv</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0h</td>
</tr>
<tr>
<td>0</td>
<td>30</td>
<td>veo_vdn_p0_pixel_hold</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0h</td>
</tr>
<tr>
<td>0</td>
<td>29</td>
<td>vdn_p0_veo_mh_dv</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0h</td>
</tr>
<tr>
<td>0</td>
<td>28</td>
<td>veo_vdn_p0_mh_hold</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0h</td>
</tr>
<tr>
<td>0</td>
<td>27</td>
<td>vdn_p0_veo_bne_luma_dv</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0h</td>
</tr>
<tr>
<td>0</td>
<td>26</td>
<td>veo_vdn_p0_bne_luma_hold</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0h</td>
</tr>
<tr>
<td>0</td>
<td>25</td>
<td>vdn_p0_veo_bne_chroma_dv</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0h</td>
</tr>
<tr>
<td>0</td>
<td>24</td>
<td>veo_vdn_p0_bne_chroma_hold</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0h</td>
</tr>
<tr>
<td>0</td>
<td>23</td>
<td>vdi_p0_veo_pixel_dv</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0h</td>
</tr>
<tr>
<td>0</td>
<td>22</td>
<td>veo_vdi_p0_pixel_hold</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0h</td>
</tr>
<tr>
<td>0</td>
<td>21</td>
<td>vdi_p0_veo_stmm_dv</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0h</td>
</tr>
</tbody>
</table>
## VEO_DVHOLD - VEO DV Hold Register

<table>
<thead>
<tr>
<th>Register Address</th>
<th>Register Name</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>veo_vdi_p0_stmm_hold</td>
<td>0h</td>
</tr>
<tr>
<td>19</td>
<td>vdi_p0_veo_fmd_dv</td>
<td>0h</td>
</tr>
<tr>
<td>18</td>
<td>veo_vdi_p0_fmd_hold</td>
<td>0h</td>
</tr>
<tr>
<td>17</td>
<td>Reserved</td>
<td>0h</td>
</tr>
<tr>
<td>16</td>
<td>Reserved</td>
<td>0h</td>
</tr>
<tr>
<td>15</td>
<td>vdn_p1_veo_pixel_dv</td>
<td>0h</td>
</tr>
<tr>
<td>14</td>
<td>veo_vdn_p1_pixel_hold</td>
<td>0h</td>
</tr>
<tr>
<td>13</td>
<td>vdn_p1_veo_mh_dv</td>
<td>0h</td>
</tr>
<tr>
<td>12</td>
<td>veo_vdn_p1_mh_hold</td>
<td>0h</td>
</tr>
<tr>
<td>11</td>
<td>vdn_p1_veo_bne_luma_dv</td>
<td>0h</td>
</tr>
<tr>
<td>10</td>
<td>veo_vdn_p1_bne_luma_hold</td>
<td>0h</td>
</tr>
<tr>
<td>9</td>
<td>vdn_p1_veo_bne_chroma_dv</td>
<td>0h</td>
</tr>
<tr>
<td>8</td>
<td>veo_vdn_p1_bne_chroma_hold</td>
<td>0h</td>
</tr>
<tr>
<td>7</td>
<td>vdi_p1_veo_pixel_dv</td>
<td>0h</td>
</tr>
<tr>
<td>6</td>
<td>veo_vdi_p1_pixel_hold</td>
<td>0h</td>
</tr>
<tr>
<td>5</td>
<td>vdi_p1_veo_stmm_dv</td>
<td>0h</td>
</tr>
<tr>
<td>4</td>
<td>veo_vdi_p1_stmm_hold</td>
<td>0h</td>
</tr>
<tr>
<td>3</td>
<td>vdi_p1_veo_fmd_dv</td>
<td>0h</td>
</tr>
</tbody>
</table>
## VEO_DVHOLD - VEO DV Hold Register

<table>
<thead>
<tr>
<th>Offset</th>
<th>Field</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td><code>veo_vdi_p1_fmd_hold</code></td>
<td><strong>0h</strong></td>
</tr>
<tr>
<td>1</td>
<td>Reserved</td>
<td><strong>0h</strong></td>
</tr>
<tr>
<td>0</td>
<td>Reserved</td>
<td><strong>0h</strong></td>
</tr>
</tbody>
</table>
VEO Previous Pipe 0 XY Register

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:30</td>
<td><strong>Reserved</strong></td>
<td></td>
<td>08858h</td>
</tr>
<tr>
<td></td>
<td>29:16</td>
<td><strong>Previous Input Pipe 0 X</strong></td>
<td>0h</td>
<td>08858h</td>
</tr>
<tr>
<td>15</td>
<td><strong>Reserved</strong></td>
<td></td>
<td></td>
<td>08958h</td>
</tr>
<tr>
<td>14:0</td>
<td><strong>Previous Input Pipe 0 Y</strong></td>
<td></td>
<td>0h</td>
<td>08958h</td>
</tr>
</tbody>
</table>
### VF Scratch Pad

- **Register Space:** MMIO: 0/2/0
- **Project:** CHV, BSW
- **Source:** RenderCS
- **Default Value:** 0x00000000
- **Access:** R/W
- **Size (in bits):** 32

<table>
<thead>
<tr>
<th>Address</th>
<th>02470h</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address</td>
<td>083A8h-083ABh</td>
</tr>
<tr>
<td>Valid Projects</td>
<td>CHV, BSW</td>
</tr>
</tbody>
</table>

#### DWord 0

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>31:16</td>
<td>Mask Bits</td>
</tr>
</tbody>
</table>

- **Format:** Mask[15:0]
- **Description:** Must be set to modify corresponding bit in Bits 15:0. (All bits implemented)

| Reserved | |
| Reserved | |
| Reserved | |
| Reserved | |
| Reserved | |
| Reserved | |

#### DWord 8

<table>
<thead>
<tr>
<th>Bits</th>
<th>End Offset Guardband Disable</th>
</tr>
</thead>
</table>

- **Project:** CHV, BSW
- **Format:** U1

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>Enable [Default]</td>
<td>When 3DPRIMITIVE.End Offset Enable is set to 1, VF will use the guardband to determine when a draw call can be pre-empted. VF will not allow pre-emption in the guardband region.</td>
</tr>
<tr>
<td>1h</td>
<td>Disable</td>
<td>When set to Disable, there will not be any preemption or GB consideration for autodraw (3DPRIMITIVE.End Offset Enable set to 1).</td>
</tr>
</tbody>
</table>

#### DWord 7

<table>
<thead>
<tr>
<th>Bits</th>
<th>Guardband Disable</th>
</tr>
</thead>
</table>

- **Format:** U1

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>Enable [Default]</td>
<td>VF will use the guardband to determine when a draw call can be pre-empted. VF will not allow pre-emption in the guardband region.</td>
</tr>
<tr>
<td>1h</td>
<td>Disable</td>
<td>VF will not use the guardband to determine when a draw call can be pre-empted. VF will allow pre-emption on any vertex in the draw call.</td>
</tr>
</tbody>
</table>
## VFSKPD - VF Scratch Pad

### 6. Reserved

### 5. TLB Prefetch Enable

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>Disable [Default]</td>
<td>The VF will generate prefetch of TLB when it is fetching sequential vertex data and four or fewer vertex buffers are valid.</td>
</tr>
<tr>
<td>1h</td>
<td>Enable</td>
<td>VF will disable prefetch of TLB entries.</td>
</tr>
</tbody>
</table>

**Project:** CHV, BSW  
**Format:** U1

### 4. Reserved

### 3. Reserved

### 2. Vertex Cache Implicit Disable Inhibit

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>[Default]</td>
<td>Allow VF to disable VS0 when Sequential index or Prim ID is a valid Element.</td>
</tr>
<tr>
<td>1h</td>
<td></td>
<td>VF never implicitly disables the vertex cache. Software must disable the VS0 Cache when required.</td>
</tr>
</tbody>
</table>

**Format:** U1

### 1. Disable Over Fetch Cache

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>[Default]</td>
<td>Cache will check for data in cache before making a request to memory</td>
</tr>
<tr>
<td>1h</td>
<td></td>
<td>Always re-fetch new data from memory.</td>
</tr>
</tbody>
</table>

**Project:** CHV, BSW

### Programming Notes

Note that the Disable Multiple Miss Read squash bit must be cleared for Disable Over Fetch Cache to be set.

### 0. Disable Multiple Miss Read squash

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>[Default]</td>
<td>Allow VF to squash reads that are to the same cacheline for vertex buffer requests.</td>
</tr>
<tr>
<td>1h</td>
<td></td>
<td>Disallow VF from squashing reads that are to the same cacheline for vertex buffer requests.</td>
</tr>
</tbody>
</table>
# VFW Credit Count Register

<table>
<thead>
<tr>
<th>VFW_CREDIT_CNT - VFW Credit Count Register</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: VideoEnhancementCS</td>
</tr>
<tr>
<td>Default Value: 0x00000002 CHV, BSW</td>
</tr>
<tr>
<td>Access: RO</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Trusted Type: 1</td>
</tr>
<tr>
<td>Address: 08810h</td>
</tr>
<tr>
<td>Address: 08910h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:8</td>
<td>Reserved</td>
</tr>
<tr>
<td>7:0</td>
<td></td>
<td>Credit Count</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The number of outstanding credits between VFW and GAV. If zero VEBOX cannot proceed due to GAV not releasing credits.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>2h</td>
<td>[Default]</td>
<td>CHV, BSW</td>
</tr>
</tbody>
</table>
VIC Virtual page Address Registers

<table>
<thead>
<tr>
<th>VICTLB_VA - VIC Virtual page Address Registers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: RenderCS</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Access: RO</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Trusted Type: 1</td>
</tr>
<tr>
<td>Address: 04900h-04903h</td>
</tr>
</tbody>
</table>

These registers are directly mapped to the current Virtual Addresses in the VICTLB (Vertex Fetch, Instruction Cache, and Command Streamer TLB.)

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:12</td>
<td><strong>Address</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: GraphicsAddress[31:12]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Page virtual address.</td>
</tr>
<tr>
<td>0</td>
<td>11:0</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
</tbody>
</table>
VIDEOBUSYCOUNTERTable

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space</td>
<td>MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits)</td>
<td>32</td>
</tr>
<tr>
<td>Address</td>
<td>138124h</td>
</tr>
</tbody>
</table>

SOXi Context Save/Restore : No
The 40-bit HW counter will wrap around. The only clear condition is CZ reset.
When 0x13_8104[15] is set to zero, bits 31:0 of the 40-bit counter are reported. When 0x13_8104[15] is set to one, its 39:8 of the 40-bit counter are reported.
The units are CZ clock cycles.
It is up to SW to periodically read this register and do a difference from the last time it was read. The primary usage for those registers for power characterization.
0x13_8104[7] controls if this register should count or if it should be gated: 0= clear, 1= count

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td>VIDEOBUSYTIME</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Render RC0 Residency Counter.</td>
</tr>
</tbody>
</table>
## Video Enhancement Mode Register

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td><strong>Mask Bits</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format:</strong> Mask[15:0]</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Description</strong>: Must be set to modify corresponding bit in Bits 15:0. (All implemented bits)</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td><strong>Execlist Enable</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default Value:</strong> 0h</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Mask:</strong> MMIO#31</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Description</strong>: When set, software can utilize the execlist registers to load a context into hardware. When this bit is clear the execlist mechanism cannot be used. The ring must be loaded via MMIO access.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Programming Notes</strong>: This bit is not intended to be changed dynamically. Changing the value of this bit while rendering is in progress will have UNDEFINED results. This bit should be changed only after a full reset and before submitting any commands to the device</td>
</tr>
<tr>
<td>14</td>
<td></td>
<td><strong>Interrupt Steering Bit</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format:</strong> U1</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Description</strong>: When set, Command Streamer sends interrupt messages to the SHIM of the ON CHIP Micro Controller through message channel. When reset, Command Streamer sends the interrupt messages to Display Engine as config writes on GAM interface.</td>
</tr>
<tr>
<td>13:10</td>
<td></td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Project:</strong> All</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format:</strong> MBZ</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td><strong>Per-Process GTT Enable</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Project:</strong> All</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format:</strong> Enable Per-Process GTT BS Mode Enable</td>
</tr>
</tbody>
</table>
## VEBOX_MODE - Video Enhancement Mode Register

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>PPGTT Disable</td>
<td>When clear, the Global GTT will be used to translate memory access from designated commands and for commands that select the PPGTT as their translation space in Basic Scheduler Mode.</td>
</tr>
<tr>
<td>1h</td>
<td>PPGTT Enable</td>
<td>When set, the PPGTT will be used to translate memory access from designated commands and for commands that select the PPGTT as their translation space.</td>
</tr>
</tbody>
</table>

### Programming Notes

This bit is used for enabling PPGTT access in Ring Buffer mode of scheduling. Privilege field in context descriptor states the same in Execlist mode of scheduling. This field should be set before programming PDP0/1/2/3 registers in order to set the PPGTT translation of memory access.

### 8:7 - Reserved

<table>
<thead>
<tr>
<th>Project:</th>
<th>CHV, BSW</th>
</tr>
</thead>
</table>

### 6:5 - 64Bit Virtual Addressing Enable

<table>
<thead>
<tr>
<th>Project:</th>
<th>CHV, BSW</th>
</tr>
</thead>
</table>

| Format: | Enable |

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>64Bit Virtual Addressing Disable</td>
<td>When clear indicates GFX operating in 32bit Virtual Addressing for PPGTT based memory access.</td>
</tr>
</tbody>
</table>

### Programming Notes

This bit is only valid when PPGTT is enabled in ring buffer mode of scheduling. Context Descriptor has a similar bit to control 64bit virtual addressing in execlist mode of scheduling. Irrespective of this field set or clear virtual addresses translated through GGTT are always 32Bit. This field should be programmed before enabling PPGTT access. When this field is not set or for GGTT virtual addresses, Graphics Address [47:32] field of any commands or register exercised by SW should be programmed to 0x0. 64 Bit Virtual Addressing is not supported on CHV, BSW and must not be set.

### 4:3 - Reserved

<table>
<thead>
<tr>
<th>Project:</th>
<th>CHV, BSW</th>
</tr>
</thead>
</table>

### 3:1 - Reserved

<table>
<thead>
<tr>
<th>Project:</th>
<th>CHV, BSW</th>
</tr>
</thead>
</table>

| Format: | MBZ |

---

1500
### VEBOX_MODE - Video Enhancement Mode Register

<table>
<thead>
<tr>
<th></th>
<th>Privilege Check Disable</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td><strong>Enable</strong></td>
</tr>
</tbody>
</table>

**Project:** CHV, BSW

This field when set, disables Privilege Violation checks on non-privileged batch buffers. When set, Privileged commands are allowed to be executed from non-privileged batch buffers.
## Video Mode Register

**MFX_MODE - Video Mode Register**

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>VideoCS, VideoCS2</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Trusted Type:</td>
<td>1</td>
</tr>
<tr>
<td>Address:</td>
<td>1229Ch</td>
</tr>
<tr>
<td>Valid Projects:</td>
<td>CHV, BSW</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:16</td>
<td>Mask Bits</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: Mask[15:0]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Must be set to modify corresponding bit in Bits 15:0. (All implemented bits)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>15</th>
<th></th>
<th>Execlist Enable</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Default Value:</td>
<td>0h</td>
</tr>
<tr>
<td></td>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td></td>
<td>Mask:</td>
<td>MMIO#31</td>
</tr>
<tr>
<td></td>
<td>When set, software can utilize the execlist registers to load a context into hardware. When this bit is clear the Execution List mechanism cannot be used. The ring must be loaded via MMIO access.</td>
<td></td>
</tr>
</tbody>
</table>

**Programming Notes**

This bit is not intended to be changed dynamically. Changing the value of this bit while rendering is in progress will have UNDEFINED results. This bit should be changed only after a full reset and before submitting any commands to the device.

<table>
<thead>
<tr>
<th>14</th>
<th></th>
<th>Interrupt Steering Bit</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td></td>
<td>Format:</td>
<td>U1</td>
</tr>
<tr>
<td></td>
<td>When set, Command Streamer sends interrupt messages to the SHIM of the ON CHIP Micro Controller through message channel. When reset, Command Streamer sends the interrupt messages to Display Engine as config writes on GAM interface.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>13:10</th>
<th></th>
<th>Reserved</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Project:</td>
<td>All</td>
</tr>
<tr>
<td></td>
<td>Format:</td>
<td>MBZ</td>
</tr>
</tbody>
</table>


# MFX_MODE - Video Mode Register

<table>
<thead>
<tr>
<th>9</th>
<th>Per-Process GTT Enable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Format:</td>
<td>Enable Per-Process GTT BS Mode Enable</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>PPGTT Disable [Default]</td>
<td>When clear, the Global GTT will be used to translate memory access from designated commands and for commands that select the PPGTT as their translation space.</td>
</tr>
<tr>
<td>1h</td>
<td>PPGTT Enable</td>
<td>When set, the PPGTT will be used to translate memory access from designated commands and for commands that select the PPGTT as their translation space.</td>
</tr>
</tbody>
</table>

## Programming Notes
This bit is used for enabling PPGTT access in Ring Buffer mode of scheduling. Privilege field in context descriptor states the same in Execlist Mode of scheduling. This field should be set before programming PDP0/1/2/3 registers in order to set the PPGTT translation of memory access.

<table>
<thead>
<tr>
<th>8</th>
<th>Reserved</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>7</th>
<th>64Bit Virtual Addressing Enable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Per-Process GTT Enable:</td>
<td>Enable</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>64Bit Virtual Addressing Disable [Default]</td>
<td>When clear indicates GFX operating in 32bit Virtual Addressing for PPGTT based memory access.</td>
</tr>
</tbody>
</table>

## Programming Notes
This bit is only valid when PPGTT is enabled in ring buffer mode of scheduling. Context Descriptor has a similar bit to control 64bit virtual addressing in execlist mode of scheduling. Irrespective of this field set or clear virtual addresses translated through GGTT are always 32Bit. This field should be programmed before enabling PPGTT access. When this field is not set or for GGTT virtual addresses, Graphics Address [47:32] field of any commands or register exercised by SW should be programmed to 0x0.

64 Bit Virtual Addressing is not supported on CHV, BSW and must not be set.

<table>
<thead>
<tr>
<th>6:5</th>
<th>Reserved</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>All</td>
</tr>
<tr>
<td>Format:</td>
<td>MBZ</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>4</th>
<th>Reserved</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
</tbody>
</table>
## MFX_MODE - Video Mode Register

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Project</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>3:1</td>
<td><strong>Reserved</strong></td>
<td>CHV, BSW</td>
<td>MBZ</td>
</tr>
<tr>
<td>0</td>
<td><strong>Privilege Check Disable</strong></td>
<td>CHV, BSW</td>
<td>Enable</td>
</tr>
</tbody>
</table>

This field when set, disables Privilege Violation checks on non-privileged batch buffers. When set, Privileged commands are allowed to be executed from non-privileged batch buffers.
## VS Invocation Counter

<table>
<thead>
<tr>
<th>VS_INVOCATION_COUNT - VS Invocation Counter</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Register Space:</strong> MMIO: 0/2/0</td>
</tr>
<tr>
<td><strong>Project:</strong> CHV, BSW</td>
</tr>
<tr>
<td><strong>Source:</strong> RenderCS</td>
</tr>
<tr>
<td><strong>Default Value:</strong> 0x00000000, 0x00000000</td>
</tr>
<tr>
<td><strong>Access:</strong> R/W</td>
</tr>
<tr>
<td><strong>Size (in bits):</strong> 64</td>
</tr>
<tr>
<td><strong>Trusted Type:</strong> 1</td>
</tr>
<tr>
<td><strong>Address:</strong> 02320h</td>
</tr>
</tbody>
</table>

This register stores the value of the vertex count shaded by VS. This register is part of the context save and restore.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>63:32</td>
<td><strong>VS Invocation Count Report UDW</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Number of vertices that are dispatched as threads by the VS stage. Updated only when Statistics Enable is set in VS_STATE (see the Vertex Shader Chapter in the 3D Volume.)</td>
</tr>
<tr>
<td>31:0</td>
<td></td>
<td><strong>VS Invocation Count Report LDW</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Number of vertices that are dispatched as threads by the VS stage. Updated only when Statistics Enable is set in VS_STATE (see the Vertex Shader Chapter in the 3D Volume.)</td>
</tr>
</tbody>
</table>
Wait For Event and Display Flip Flags Register

SYNC_FLIP_STATUS - Wait For Event and Display Flip Flags Register

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
</tbody>
</table>

Address: 022D0h
Name: RCS Wait For Event and Display Flip Flags Register
ShortName: RCS_SYNC_FLIP_STATUS
Valid Projects: CHV, BSW
Address: 122D0h-122D3h
Name: Wait For Event and Display Flip Flags Register
ShortName: SYNC_FLIP_STATUS_VCSUNIT0
Address: 1A2D0h-1A2D3h
Name: Wait For Event and Display Flip Flags Register
ShortName: SYNC_FLIP_STATUS_VECSUNIT
Address: 1C2D0h-1C2D3h
Name: Wait For Event and Display Flip Flags Register
ShortName: SYNC_FLIP_STATUS_VCSUNIT1
Address: 222D0h
Name: BCS Wait For Event and Display Flip Flags Register
ShortName: BCS_SYNC_FLIP_STATUS
Valid Projects: CHV, BSW

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

**Programming Notes**

**Programming Restriction:** This register should NEVER be programmed by SW, this is for HW internal use only.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: MBZ</td>
</tr>
<tr>
<td>30</td>
<td></td>
<td><strong>Display Plane A Asynchronous Display Flip Pending</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format: Enable</td>
</tr>
</tbody>
</table>

This register is the saved value of what wait for events are still valid. This register is part of context save and restore for RC6 feature.
## SYNC_FLIP_STATUS - Wait For Event and Display Flip Flags Register

This field enables a wait for the duration of a Display Plane A "Flip Pending" condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition (in the Device Programming Interface chapter of MI Functions).

### 29 Display Plane A Synchronous Flip Display Pending

<table>
<thead>
<tr>
<th>Format</th>
<th>Enable</th>
</tr>
</thead>
</table>

This field enables a wait for the duration of a Display Plane A "Flip Pending" condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition (in the Device Programming Interface chapter of MI Functions).

### 28 Display Sprite A Synchronous Flip Display Pending

<table>
<thead>
<tr>
<th>Format</th>
<th>Enable</th>
</tr>
</thead>
</table>

This field enables a wait for the duration of a Display Sprite A "Flip Pending" condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition in the Device Programming Interface chapter of MI Functions.

### 27 Reserved

<table>
<thead>
<tr>
<th>Format</th>
<th>MBZ</th>
</tr>
</thead>
</table>

### 26 Display Plane B Asynchronous Display Flip Pending

<table>
<thead>
<tr>
<th>Format</th>
<th>Enable</th>
</tr>
</thead>
</table>

This field enables a wait for the duration of a Display Plane B "Flip Pending" condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition (in the Device Programming Interface chapter of MI Functions).

### 25 Display Plane B Synchronous Flip Display Pending

<table>
<thead>
<tr>
<th>Format</th>
<th>Enable</th>
</tr>
</thead>
</table>

This field enables a wait for the duration of a Display Plane B Flip Pending condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition (in the Device Programming Interface chapter of MI Functions).

### 24 Display Sprite B Synchronous Flip Display Pending

<table>
<thead>
<tr>
<th>Format</th>
<th>Enable</th>
</tr>
</thead>
</table>

This field enables a wait for the duration of a Display Sprite B Flip Pending condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition in the Device Programming Interface chapter of MI Functions.
### SYNC_FLIP_STATUS - Wait For Event and Display Flip Flags Register

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
<th>Source</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>23</td>
<td>Reserved</td>
<td>BlitterCS</td>
<td>MBZ</td>
</tr>
<tr>
<td>23</td>
<td>Display Plane A Asynchronous Performance Flip Pending Wait Enable</td>
<td>RenderCS</td>
<td>Enable</td>
</tr>
<tr>
<td></td>
<td>This field enables a wait for the duration of a Display Plane A Flip Pending condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition in the Device Programming Interface chapter of MI Functions.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>Display Plane A Asynchronous Flip Pending Wait Enable</td>
<td></td>
<td>Enable</td>
</tr>
<tr>
<td></td>
<td>This field enables a wait for the duration of a Display Plane A Flip Pending condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition in the Device Programming Interface chapter of MI Functions.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>Display Plane A Synchronous Flip Pending Wait Enable</td>
<td></td>
<td>Enable</td>
</tr>
<tr>
<td></td>
<td>This field enables a wait for the duration of a Display Plane A Flip Pending condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition in the Device Programming Interface chapter of MI Functions.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>Display Sprite A Synchronous Flip Pending Wait Enable</td>
<td></td>
<td>Enable</td>
</tr>
<tr>
<td></td>
<td>This field enables a wait for the duration of a Display Sprite A Flip Pending condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition in the Device Programming Interface chapter of MI Functions.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>Reserved</td>
<td></td>
<td>MBZ</td>
</tr>
<tr>
<td>18</td>
<td>Display Pipe A Scan Line Wait Enable</td>
<td></td>
<td>Enable</td>
</tr>
<tr>
<td></td>
<td>This field enables a wait while a Display Pipe A Scan Line condition exists. This condition is defined as the the start of the scan line specified in the Pipe A Display Scan Line Count Range Compare Register. See Scan Line Event in the Device Programming Interface chapter of MI Functions.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### SYNC_FLIP_STATUS - Wait For Event and Display Flip Flags Register

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
<th>Format</th>
<th>Source</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td><strong>Display Pipe A Vertical Blank Wait Enable</strong></td>
<td>Enable</td>
<td></td>
<td>See Vertical Blank Event (See Programming Interface)</td>
</tr>
<tr>
<td>16</td>
<td><strong>Reserved</strong></td>
<td>MBZ</td>
<td>CHV, BSW</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td><strong>Display Plane B Asynchronous Performance Flip Pending Wait Enable</strong></td>
<td>MBZ</td>
<td>BlitterCS</td>
<td>See Display Flip Pending Condition (in the Device Programming Interface chapter of MI Functions)</td>
</tr>
<tr>
<td>14</td>
<td><strong>Display Plane B Asynchronous Flip Pending Wait Enable</strong></td>
<td>Enable</td>
<td>RenderCS</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td><strong>Display Plane B Synchronous Flip Pending Wait Enable</strong></td>
<td>Enable</td>
<td></td>
<td>See Display Flip Pending Condition (in the Device Programming Interface chapter of MI Functions)</td>
</tr>
<tr>
<td>12</td>
<td><strong>Display Sprite B Synchronous Flip Pending Wait Enable</strong></td>
<td>Enable</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## SYNC_FLIP_STATUS - Wait For Event and Display Flip Flags Register

This field enables a wait for the duration of a Display Sprite B Flip Pending condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition in the Device Programming Interface chapter of MI Functions.

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>Reserved</td>
<td>MBZ</td>
</tr>
<tr>
<td>10</td>
<td>Display Pipe B Scan Line Wait Enable</td>
<td>Enable</td>
</tr>
<tr>
<td>9</td>
<td>Display Pipe B Vertical Blank Wait Enable</td>
<td>Enable</td>
</tr>
<tr>
<td>8</td>
<td>Reserved</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>7:5</td>
<td>Reserved</td>
<td>MBZ</td>
</tr>
<tr>
<td>4:0</td>
<td>Reserved</td>
<td>CHV, BSW</td>
</tr>
</tbody>
</table>

- **Sync Flip Status**
  - This field enables a wait for the duration of a Display Sprite B Flip Pending condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition in the Device Programming Interface chapter of MI Functions.

- **Display Pipe B Scan Line Wait Enable**
  - Format: Enable
  - This field enables a wait while a Display Pipe B Scan Line condition exists. This condition is defined as the the start of the scan line specified in the Pipe B Display Scan Line Count Range Compare Register. See Scan Line Event in the Device Programming Interface chapter of MI Functions.

- **Display Pipe B Vertical Blank Wait Enable**
  - Format: Enable
  - This field enables a wait until the next Display Pipe B Vertical Blank event occurs. This event is defined as the start of the next Display Pipe B vertical blank period. Note that this can cause a wait for up to an entire refresh period. See Vertical Blank Event (See Programming Interface).
**Command Reference: Registers**

---

## Wait For Event and Display Flip Flags Register 1

### `SYNC_FLIP_STATUS_1` - Wait For Event and Display Flip Flags Register 1

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Access:</td>
<td>R/W</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
</tbody>
</table>

**Address:** 022D4h

**Name:** RCS Wait For Event and Display Flip Flags Register 1

**ShortName:** RCS_SYNC_FLIP_STATUS_1

**Address:** 122D4h-122D7h

**Name:** Wait For Event and Display Flip Flags Register 1

**ShortName:** SYNC_FLIP_STATUS_1_VCSUNIT0

**Address:** 1A2D4h-1A2D7h

**Name:** Wait For Event and Display Flip Flags Register 1

**ShortName:** SYNC_FLIP_STATUS_1_VECSUNIT

**Address:** 1C2D4h-1C2D7h

**Name:** Wait For Event and Display Flip Flags Register 1

**ShortName:** SYNC_FLIP_STATUS_1_VCSUNIT1

**Address:** 222D4h

**Name:** BCS Wait For Event and Display Flip Flags Register 1

**ShortName:** BCS_SYNC_FLIP_STATUS_1

**Valid Projects:** CHV, BSW

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

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:27</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td>26</td>
<td></td>
<td><strong>Display Sprite C3 Synchronous Flip Pending Wait Enable</strong></td>
</tr>
</tbody>
</table>

- **Project:** CHV, BSW
- **Format:** Enable

This field enables a wait for the duration of a Display Sprite C3 Flip Pending condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition in the Device Programming Interface chapter of MI Functions.
### SYNC_FLIP_STATUS_1 - Wait For Event and Display Flip Flags

#### Register 1

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
<th>Project</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>Display Sprite C3 Synchronous Flip Display Pending</td>
<td>CHV, BSW</td>
<td>Enable</td>
</tr>
<tr>
<td></td>
<td>This field enables a wait for the duration of a Display Sprite C3 Flip Pending condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition in the Device Programming Interface chapter of MI Functions.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>Display Sprite B3 Synchronous Flip Pending Wait Enable</td>
<td>CHV, BSW</td>
<td>Enable</td>
</tr>
<tr>
<td></td>
<td>This field enables a wait for the duration of a Display Sprite B3 Flip Pending condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition in the Device Programming Interface chapter of MI Functions.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>Display Sprite B3 Synchronous Flip Display Pending</td>
<td>CHV, BSW</td>
<td>Enable</td>
</tr>
<tr>
<td></td>
<td>This field enables a wait for the duration of a Display Sprite B3 Flip Pending condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition in the Device Programming Interface chapter of MI Functions.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>Display Sprite A3 Synchronous Flip Pending Wait Enable</td>
<td>CHV, BSW</td>
<td>Enable</td>
</tr>
<tr>
<td></td>
<td>This field enables a wait for the duration of a Display Sprite A3 Flip Pending condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition in the Device Programming Interface chapter of MI Functions.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>Display Sprite A3 Synchronous Flip Display Pending</td>
<td>CHV, BSW</td>
<td>Enable</td>
</tr>
<tr>
<td></td>
<td>This field enables a wait for the duration of a Display Sprite A3 Flip Pending condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition in the Device Programming Interface chapter of MI Functions.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## SYNC_FLIP_STATUS_1 - Wait For Event and Display Flip Flags

### Register 1

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>Display Sprite C2 Synchronous Flip Pending Wait Enable</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>20</strong></td>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td></td>
<td>Format:</td>
<td>Enable</td>
</tr>
</tbody>
</table>

This field enables a wait for the duration of a Display Sprite C2 Flip Pending condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition in the Device Programming Interface chapter of MI Functions.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>Display Sprite C2 Synchronous Flip Display Pending</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>19</strong></td>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td></td>
<td>Format:</td>
<td>Enable</td>
</tr>
</tbody>
</table>

This field enables a wait for the duration of a Display Sprite C2 Flip Pending condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition in the Device Programming Interface chapter of MI Functions.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>Display Sprite B2 Synchronous Flip Pending Wait Enable</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>18</strong></td>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td></td>
<td>Format:</td>
<td>Enable</td>
</tr>
</tbody>
</table>

This field enables a wait for the duration of a Display Sprite B2 Flip Pending condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition in the Device Programming Interface chapter of MI Functions.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>Display Sprite B2 Synchronous Flip Display Pending</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>17</strong></td>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td></td>
<td>Format:</td>
<td>Enable</td>
</tr>
</tbody>
</table>

This field enables a wait for the duration of a Display Sprite B2 Flip Pending condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition in the Device Programming Interface chapter of MI Functions.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>Display Sprite A2 Synchronous Flip Pending Wait Enable</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>16</strong></td>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td></td>
<td>Format:</td>
<td>Enable</td>
</tr>
</tbody>
</table>

This field enables a wait for the duration of a Display Sprite A2 Flip Pending condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition in the Device Programming Interface chapter of MI Functions.
### SYNC_FLIP_STATUS_1 - Wait For Event and Display Flip Flags

#### Register 1

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
<th>Project</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td><strong>Display Sprite A2 Synchronous Flip Display Pending</strong></td>
<td>CHV, BSW</td>
<td>Enable</td>
</tr>
<tr>
<td></td>
<td>This field enables a wait for the duration of a Display Sprite A2 Flip Pending condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition in the Device Programming Interface chapter of MI Functions.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td><strong>Display Plane C Scan Line Event Pending</strong></td>
<td>CHV, BSW</td>
<td>Enable</td>
</tr>
<tr>
<td></td>
<td>This field indicates scan line event operation is pending from Display Plane-C. This field gets set when MI_SCANLINE_INCL/EXCL command is parsed for Display Plane-C and gets reset on scan line event completion for Display Plane-C.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td><strong>Display Plane B Scan Line Event Pending</strong></td>
<td>CHV, BSW</td>
<td>Enable</td>
</tr>
<tr>
<td></td>
<td>This field indicates scan line event operation is pending from Display Plane-B. This field gets set when MI_SCANLINE_INCL/EXCL command is parsed for Display Plane-B and gets reset on scan line event completion for Display Plane-B.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td><strong>Display Plane A Scan Line Event Pending</strong></td>
<td>CHV, BSW</td>
<td>Enable</td>
</tr>
<tr>
<td></td>
<td>This field indicates scan line event operation is pending from Display Plane-A. This field gets set when MI_SCANLINE_INCL/EXCL command is parsed for Display Plane-A and gets reset on scan line event completion for Display Plane-A.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td><strong>Reserved</strong></td>
<td>CHV, BSW</td>
<td>MBZ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td><strong>Display Plane C Asynchronous Display Flip Pending</strong></td>
<td></td>
<td>Enable</td>
</tr>
<tr>
<td></td>
<td>This field enables a wait for the duration of a Display Plane C Flip Pending condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition in the Device Programming Interface chapter of MI Functions.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### SYNC_FLIP_STATUS_1 - Wait For Event and Display Flip Flags

#### Register 1

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
<th>Format</th>
<th>Source</th>
<th>MBZ</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>Display Plane C Synchronous Flip Display Pending</td>
<td>Enable</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>This field enables a wait for the duration of a Display Plane C Flip Pending condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition (in the Device Programming Interface chapter of MI Functions).</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Display Sprite C Synchronous Flip Display Pending</td>
<td>Enable</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>This field enables a wait for the duration of a Display Sprite C Flip Pending condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition (in the Device Programming Interface chapter of MI Functions).</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Reserved</td>
<td>BlitterCS</td>
<td>MBZ</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Source: BlitterCS</td>
<td>Format: MBZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Display Plane C Asynchronous Performance Flip Pending Wait Enable</td>
<td>Enable</td>
<td>RenderCS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>This field enables a wait for the duration of a Display Plane C Flip Pending condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition (in the Device Programming Interface chapter of MI Functions).</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Display Plane C Asynchronous Flip Pending Wait Enable</td>
<td>Enable</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>This field enables a wait for the duration of a Display Plane C &quot;Flip Pending&quot; condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition (in the Device Programming Interface chapter of MI Functions).</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Display Plane C Synchronous Flip Pending Wait Enable</td>
<td>Enable</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>This field enables a wait for the duration of a Display Plane C Flip Pending condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition (in the Device Programming Interface chapter of MI Functions).</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## SYNC_FLIP_STATUS_1 - Wait For Event and Display Flip Flags

### Register 1

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Display Sprite C Synchronous Flip Pending Wait Enable</td>
<td>Enable</td>
<td>This field enables a wait for the duration of a Display Sprite C Flip Pending condition. If a flip request is pending, the parser will wait until the flip operation has completed (i.e., the new front buffer address has now been loaded into the active front buffer registers). See Display Flip Pending Condition in the Device Programming Interface chapter of MI Functions.</td>
</tr>
<tr>
<td>3</td>
<td>Reserved</td>
<td>MBZ</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Display Pipe C Scan Line Wait Enable</td>
<td>Enable</td>
<td>This field enables a wait while a Display Pipe C Scan Line condition exists. This condition is defined as the the start of the scan line specified in the Pipe C Display Scan Line Count Range Compare Register. See Scan Line Event in the Device Programming Interface chapter of MI Functions.</td>
</tr>
<tr>
<td>1</td>
<td>Display Pipe C Vertical Blank Wait Enable</td>
<td>Enable</td>
<td>This field enables a wait until the next Display Pipe C Vertical Blank event occurs. This event is defined as the start of the next Display Pipe C vertical blank period. Note that this can cause a wait for up to an entire refresh period. See Vertical Blank Event (See Programming Interface).</td>
</tr>
<tr>
<td>0</td>
<td>Reserved</td>
<td>CHV, BSW</td>
<td></td>
</tr>
</tbody>
</table>

Project: CHV, BSW

Format: MBZ
## WF_REG - Walkers Fault Register

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:1</td>
<td>Walkers Fault Register</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000000000000000000000000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>All bits are only valid with bit[0]=1.</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td>Valid Bit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This bit indicates that the first fault for this engine has been recorded. It can only be cleared by SW, which also clears the other fields.</td>
</tr>
</tbody>
</table>
# WD_WNIC_MSG_ADDR

<table>
<thead>
<tr>
<th>WD_WNIC_MSG_ADDR - WD_WNIC_MSG_ADDR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 6E530h</td>
</tr>
</tbody>
</table>

WNIC Message Address. This is a register within Gunit (CZ domain). Address uses the address as shown.

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>Reserved</strong></td>
</tr>
</tbody>
</table>
WGBOX State Arbitration Priority Control

**APC - WGBOX State Arbitration Priority Control**

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:9</td>
<td><strong>Reserved</strong></td>
</tr>
</tbody>
</table>

Format: MBZ
# WIDI LRA 0

## WIDI_LRA_0 - WIDI LRA 0

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:30</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00b</td>
</tr>
<tr>
<td>29:24</td>
<td><strong>WIDI LRA1 Max</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default Value: 110111b</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Maximum value of programmable LRA1.</td>
<td></td>
</tr>
<tr>
<td>23:22</td>
<td><strong>Reserved</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default Value: 00b</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access: RO</td>
<td></td>
</tr>
<tr>
<td>21:16</td>
<td><strong>WIDI LRA1 Min</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default Value: 100000b</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Minimum value of programmable LRA1.</td>
<td></td>
</tr>
<tr>
<td>15:14</td>
<td><strong>Reserved</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default Value: 00b</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access: RO</td>
<td></td>
</tr>
<tr>
<td>13:8</td>
<td><strong>WIDI LRA0 Max</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default Value: 011111b</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Maximum value of programmable LRA0.</td>
<td></td>
</tr>
<tr>
<td>7:6</td>
<td><strong>Reserved</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default Value: 00b</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access: RO</td>
<td></td>
</tr>
<tr>
<td>5:0</td>
<td><strong>WIDILRA0 Min</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default Value: 000000b</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Access: R/W</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Minimum value of programmable LRA0.</td>
<td></td>
</tr>
</tbody>
</table>

- **Register Space:** MMIO: 0/2/0
- **Project:** CHV, BSW
- **Source:** PRM
- **Default Value:** 0x37201F00
- **Size (in bits):** 32
- **Address:** 04A90h

---

**Note:**

- WIDI_LRA_0 is a register used in the context of the Intel architecture to manage LRA (Load/Store) operations. The register is part of the MMIO space and is accessible via Dword addresses.
- The register contains fields for setting and retrieving maximum and minimum values for programmable LRA operations, which can be useful for optimizing memory access patterns in the system.
- The Dword and Bit fields provide a detailed view of the register structure, allowing for precise configuration of the LRA parameters.

---

**Source:** Intel Corporation

**License:**

This content is licensed under the Creative Commons Attribution-NonCommercial 4.0 International License. You are free to share and adapt this content as long as you give appropriate credit to Intel Corporation and do not use the content for commercial purposes.
## WIDI LRA 1

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:30</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>29:28</td>
<td>VMX</td>
<td>Which LRA should VMX use.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td>27:26</td>
<td>BSP</td>
<td>Which LRA should BSP use.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td>25:24</td>
<td>IME</td>
<td>Which LRA should IME/WRS use.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 01b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td>23:14</td>
<td>Reserved</td>
<td>Which LRA should IME/WRS use.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0000000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>13:8</td>
<td>Reserved</td>
<td>Which LRA should IME/WRS use.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>7:6</td>
<td>Reserved</td>
<td>Which LRA should IME/WRS use.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td>5:0</td>
<td>Reserved</td>
<td>Which LRA should IME/WRS use.</td>
</tr>
</tbody>
</table>

Register Space: MMIO: 0/2/0
Project: CHV, BSW
Source: PRM
Default Value: 0x01000000
Size (in bits): 32
Address: 04A94h

### WIDI TLB Control Register

#### WTCR - WIDI TLB Control Register

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>04278h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Default Value:</th>
<th>Access:</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:1</td>
<td>Reserved</td>
<td>0000000000000000000000000000000b</td>
<td>RO</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>Invalidate TLBs on the corresponding Engine</td>
<td>0b</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SW writes 1 to invalidate the TLBs for the associated engine and HW clears the bit when invalidation is complete. To ensure proper invalidation of the TLBs, SW has to ensure the corresponding engine's HW pipeline is flushed and cleared from all its memory accesses. Otherwise HW cannot guarantee the proper invalidation for TLBs. This bit is self clear.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### WRID_VALID_REG0

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>WRID_VALID_REG0</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
</tbody>
</table>

This register is for WRID Comparison usage. RO register with IA Access Type on DEV reset.
wrtp_wrid_valid_vector[31:0] There are 96 write buffer. Each bit indicate the buffer is valid if set.
Divide into 3 registers to accommodate all 96 deep
WRID_VALID_REG1

<table>
<thead>
<tr>
<th>WRID_VALID_REG1 - WRID_VALID_REG1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Space: MMIO: 0/2/0</td>
</tr>
<tr>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td>Source: PRM</td>
</tr>
<tr>
<td>Default Value: 0x00000000</td>
</tr>
<tr>
<td>Size (in bits): 32</td>
</tr>
<tr>
<td>Address: 04074h</td>
</tr>
<tr>
<td>DWord</td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

This register is for WRID Comparison usage. RO register with IA Access Type on DEV reset. wrdp_wrid_valid_vector[63:32] There are 96 write buffer. Each bit indicate the buffer is valid if set. Divide into 3 registers to accommodate all 96 deep.
# WRID VALID REG2

<table>
<thead>
<tr>
<th>Register Space:</th>
<th>MMIO: 0/2/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project:</td>
<td>CHV, BSW</td>
</tr>
<tr>
<td>Source:</td>
<td>PRM</td>
</tr>
<tr>
<td>Default Value:</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Size (in bits):</td>
<td>32</td>
</tr>
<tr>
<td>Address:</td>
<td>04078h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:0</td>
<td><strong>WRID VALID REG2</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00000000h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
</tbody>
</table>

This register is for WRID Comparison usage. RO register with IA Access Type on DEV reset. wrdp_wrid_valid_vector[95:64] There are 96 write buffer. Each bit indicate the buffer is valid if set. Divide into 3 registers to accommodate all 96 deep.
## WR_WATERMARK - Write Watermark

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:20</td>
<td><strong>Extra Bits</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 000000000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
</tbody>
</table>

| 19    | Watermark Timeout Enable |
|       | Default Value: 1b |
|       | Access: R/W |

| 18:8  | Watermark Timeout |
|       | Default Value: 11111111110b |
|       | Access: R/W |

Number of clocks that the write pipe queue is allowed to keep a ready write cycle, without reads or writes to the queue. Once this value is met, and if the feature is enabled, the watermark is considered reached, and all pending write requests are issued.

| 7     | Watermark Enable |
|       | Default Value: 1b |
|       | Access: R/W |

Enable Write Request Grouping

| 6:0   | **High Watermark** |
|       | Default Value: 0100100b |
|       | Access: R/W |

This is the number of write requests to be collected before initiating a write burst. Once a burst is initiated, it continues until all the available writes are requested.
# ZTLB LRA 0

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31:29</td>
<td><strong>Reserved</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project: CHV, BSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: RO</td>
</tr>
<tr>
<td></td>
<td>28:27</td>
<td><strong>STC LRA</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Which LRA should STC use.</td>
</tr>
<tr>
<td></td>
<td>26:18</td>
<td><strong>ZTLB LRA1 Min</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 00100000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Minimum value of programmable LRA1.</td>
</tr>
<tr>
<td></td>
<td>17:9</td>
<td><strong>ZTLB LRA0 Max</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0001111111b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Maximum value of programmable LRA0.</td>
</tr>
<tr>
<td></td>
<td>8:0</td>
<td><strong>ZTLB LRA0 Min</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value: 0000000000b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access: R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Minimum value of programmable LRA0.</td>
</tr>
</tbody>
</table>
### ZTLB LRA 1

**Register Space:** MMIO: 0/2/0  
**Project:** CHV, BSW  
**Source:** PRM  
**Default Value:** 0x33FD80BF CHV, BSW  
**Size (in bits):** 32  
**Address:** 04A34h

<table>
<thead>
<tr>
<th>DWord</th>
<th>Bit</th>
<th>Description</th>
<th>Value</th>
<th>Name</th>
<th>Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31</td>
<td>Reserved</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value:</td>
<td>0b</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access:</td>
<td>RO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>30:29</td>
<td></td>
<td>HIZ LRA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value:</td>
<td>01b</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access:</td>
<td>R/W</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Which LRA should HIZ use.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>28:27</td>
<td></td>
<td>RCZ LRA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value:</td>
<td>10b</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access:</td>
<td>R/W</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Which LRA should RCZ use.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>26:18</td>
<td></td>
<td>ZTLB LRA2 Max</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access:</td>
<td>R/W</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Maximum value of programmable LRA2.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Value</td>
<td>Name</td>
<td>Project</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>011111111b</td>
<td>[Default]</td>
<td>CHV, BSW</td>
<td></td>
</tr>
<tr>
<td>17:9</td>
<td></td>
<td>ZTLB LRA2 Min</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value:</td>
<td>011000000b</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access:</td>
<td>R/W</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Minimum value of programmable LRA2.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8:0</td>
<td></td>
<td>ZTLB LRA1 Max</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default Value:</td>
<td>010111111b</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access:</td>
<td>R/W</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Maximum value of programmable LRA1.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>