# BIOS and Kernel Developer's Guide (BKDG) For AMD Family 12h Processors

# Table of Contents

**1 Overview . . . 16**
1.1 Intended Audience . . . 16
1.2 Reference Documents . . . 16
1.3 Conventions . . . 16
1.3.1 Numbering . . . 16
1.3.2 Arithmetic And Logical Operators . . . 17
1.4 Definitions . . . 17

**2 Functional Description . . . 22**
2.1 Processor Overview . . . 22
2.2 System Overview . . . 22
2.3 Processor Initialization . . . 23
2.3.1 BSP initialization . . . 23
2.3.2 AP initialization . . . 23
2.3.3 Cache Initialization For General Storage During Boot . . . 24
2.3.4 BIOS Requirements For 64-Bit Operation . . . 26
2.4 Processor Core . . . 26
2.4.1 Virtual Address Space . . . 26
2.4.2 Processor Cores and Downcoring . . . 26
2.4.3 Access Type Determination . . . 26
2.4.4 System Address Map . . . 26
2.4.4.1 Memory Access to the Physical Address Space . . . 26
2.4.4.1.1 Determining The Cache Attribute . . . 26
2.4.4.1.2 Determining The Access Destination for CPU Accesses . . . 27
2.4.5 Timers . . . 27
2.4.6 Interrupts . . . 27
2.4.6.1 Local APIC . . . 27
2.4.6.1.1 ApicId Enumeration Requirements . . . 28
2.4.6.1.2 Physical Destination Mode . . . 28
2.4.6.1.3 Logical Destination Mode . . . 29
2.4.6.1.4 Interrupt Delivery . . . 29
2.4.6.1.5 Vectored Interrupt Handling . . . 29
2.4.6.1.6 Interrupt Masking . . . 29
2.4.6.1.7 Spurious Interrupts . . . 30
2.4.6.1.8 Spurious Interrupts Caused by Timer Tick Interrupt . . . 30
2.4.6.1.9 Lowest-Priority Interrupt Arbitration . . . 30
2.4.6.1.10 Inter-Processor Interrupts . . . 31
2.4.6.1.11 APIC Timer Operation . . . 31
2.4.6.1.12 Generalized Local Vector Table . . . 31
2.4.6.1.13 State at Reset . . . 31
2.4.6.2 System Management Mode (SMM) . . . 31
2.4.6.2.1 SMM Overview . . . 31
2.4.6.2.2 Operating Mode and Default Register Values . . . 32
2.4.6.2.3 SMI Sources And Delivery . . . 32
2.4.6.2.4 SMM Initial State . . . 33
2.4.6.2.5 SMM Save State . . . 33
2.4.6.2.6 Exceptions and Interrupts in SMM . . . 39

2.4.6.2.7 The Protected ASeg and TSeg Areas . . . 39
2.4.6.2.8 SMM Special Cycles . . . 39
2.4.6.2.9 Locking SMM . . . 39
2.4.7 Secure Virtual Machine Mode (SVM) . . . 39
2.4.7.1 BIOS support for SVM Disable . . . 40
2.4.8 CPUID Instruction . . . 40
2.4.8.1 Multi-Core Support . . . 40
2.5 Power Management . . . 40
2.5.1 Processor Power Planes And Voltage Control . . . 41
2.5.1.1 Internal VID Registers . . . 41
2.5.1.1.1 VID Encodings . . . 41
2.5.1.1.2 MinVid and MaxVid Check . . . 41
2.5.1.2 Serial VID Interface . . . 41
2.5.1.3 BIOS Requirements for Power Plane Initialization . . . 41
2.5.1.4 Low Power Features . . . 42
2.5.1.4.1 PSI_L Bit . . . 42
2.5.1.4.1.1 BIOS Requirements for PSI_L . . . 42
2.5.1.4.2 Alternate Low Power Voltages . . . 43
2.5.1.4.3 Power Gating . . . 44
2.5.1.5 Voltage Transitions . . . 44
2.5.1.5.1 Hardware-Initiated Voltage Transitions . . . 44
2.5.1.5.2 Software-Initiated Voltage Transitions . . . 44
2.5.2 Frequency and Voltage Domain Dependencies . . . 45
2.5.2.1 Dependencies Between Cores . . . 45
2.5.2.2 Dependencies Between Subcomponents on VDDNB . . . 45
2.5.3 CPU Power Management . . . 46
2.5.3.1 Core P-states . . . 46
2.5.3.1.1 Core Performance Boost (CPB) . . . 46
2.5.3.1.2 Core P-state Naming and Numbering . . . 46
2.5.3.1.2.1 Software P-state Numbering . . . 47
2.5.3.1.2.2 Hardware P-state Numbering . . . 47
2.5.3.1.3 Core P-state Control . . . 47
2.5.3.1.4 Core P-state Visibility . . . 48
2.5.3.1.5 Core P-state Limits . . . 48
2.5.3.1.6 Core P-state Transition Behavior . . . 49
2.5.3.1.7 BIOS Requirements for Core P-State Initialization and Transitions . . . 49
2.5.3.1.8 Processor-Systemboard Power Delivery Check . . . 50
2.5.3.1.9 ACPI Processor P-State Objects . . . 51
2.5.3.1.9.1 _PCT (Performance Control) . . . 51
2.5.3.1.9.2 _PSS (Performance Supported States) . . . 52
2.5.3.1.9.3 _PPC (Performance Present Capabilities) . . . 53
2.5.3.1.9.4 _PSD (P-State Dependency) . . . 53
2.5.3.1.9.5 Fixed ACPI Description Table (FADT) Entries . . . 53
2.5.3.1.10 XPSS (Microsoft® Extended PSS) Object . . . 53
2.5.3.2 C-states . . . 53
2.5.3.2.1 C-state Names and Numbers . . . 53
2.5.3.2.2 C-state Request Interface . . . 53
2.5.3.2.3 C-state Actions . . . 56
2.5.3.2.3.1 Core C1 (CC1) State . . . 56
2.5.3.2.3.2 Core C6 (CC6) State . . . 56

2.5.3.2.3.3 Package C1 (PC1) State . . . 56
2.5.3.2.3.4 Package C6 (PC6) State . . . 56
2.5.3.2.4 C-state Request Monitors . . . 57
2.5.3.2.4.1 DMA Tracking . . . 57
2.5.3.2.4.2 FCH Messaging . . . 57
2.5.3.2.4.3 Interrupt Monitors . . . 57
2.5.3.2.4.4 Residency Monitors . . . 58
2.5.3.2.4.5 C-state Monitor Masking . . . 59
2.5.3.2.5 C-states and Probe Requests . . . 59
2.5.3.2.6 Exiting C-states . . . 59
2.5.3.2.7 C-state initiated P-state Changes . . . 60
2.5.3.2.7.1 Auto-Pmin . . . 60
2.5.3.2.7.2 Exiting PC6 . . . 60
2.5.3.2.8 ACPI Processor C-state Objects . . . 61
2.5.3.2.8.1 _CST (C-state) . . . 61
2.5.3.2.8.2 _CRS . . . 61
2.5.3.2.8.3 Fixed ACPI Description Table (FADT) Entries . . . 62
2.5.3.2.9 BIOS Requirements for C-state Initialization . . . 62
2.5.3.3 Effective Frequency . . . 62
2.5.4 Northbridge Power Management . . . 63
2.5.4.1 NB P-states . . . 63
2.5.4.1.1 BIOS Requirements for NB P-state Initialization During DRAM Training . . . 65
2.5.4.1.2 System BIOS Requirements for NB P-state Operation During POST . . . 66
2.5.4.1.3 Software Controlled NB P-states . . . 66
2.5.4.2 NB Clock Ramping . . . 66
2.5.4.3 NB Clock Gating . . . 66
2.5.5 Root Complex Power Management . . . 67
2.5.5.1 Dynamic Power Management (DPM) . . . 67
2.5.5.1.1 Activity Monitor . . . 67
2.5.5.1.2 SCLK DPM . . . 68
2.5.5.1.3 LCLK DPM . . . 68
2.5.5.1.3.1 LCLK DPM Using Root Complex Activity . . . 69
2.5.5.1.3.2 LCLK DPM Using PCIe Bandwidth and GFX DMA Activity . . . 69
2.5.5.1.3.3 BIOS Initialization for LCLK DPM Using Root Complex Activity . . . 70
2.5.5.2 LCLK Clock Ramping (Deep Sleep) . . . 71
2.5.5.2.1 LCLK Deep Sleep . . . 71
2.5.5.2.1.1 BIOS Initialization for LCLK Deep Sleep . . . 71
2.5.5.3 GPU and Root Complex Clock Gating . . . 71
2.5.5.4 GPU and Root Complex Power Gating . . . 72
2.5.5.4.1 GPU Power Gating . . . 72
2.5.5.4.1.1 Software Requirements for GPU Power Gating . . . 72
2.5.5.4.2 GMC Power Gating . . . 73
2.5.5.4.2.1 BIOS Initialization for GMC Power Gating . . . 73
2.5.5.4.3 UVD Power Gating . . . 74
2.5.5.4.3.1 Software Requirements for UVD Power Gating . . . 74
2.5.5.4.4 Gfx Link Core Power Gating . . . 75
2.5.5.4.4.1 Software Initialization for Gfx Link Core Power Gating . . . 75
2.5.5.5 PCIe Speed Power Policy (PSPP) . . . 76
2.5.5.5.1 ALIB . . . 76
2.5.5.5.2 PSPP Power Policies . . . 76

2.5.6 DRAM Power Management . . . 77
2.5.6.1 DRAM Self-Refresh . . . 77
2.5.6.1.1 Stutter Mode . . . 77
2.5.6.1.1.1 System BIOS Requirements for Stutter Mode Operation During POST . . . 77
2.5.6.2 EVENT_L . . . 78
2.5.7 System Power Management . . . 78
2.5.7.1 S-states . . . 78
2.5.7.1.1 ACPI Suspend to RAM State (S3) . . . 78
2.6 Performance Monitoring . . . 78
2.6.1 Performance Monitor Counters . . . 78
2.6.2 Instruction Based Sampling (IBS) . . . 79
2.7 Configuration Space . . . 79
2.7.1 MMIO Configuration Coding Requirements . . . 80
2.7.2 MMIO Configuration Ordering . . . 80
2.7.3 Processor Configuration Space . . . 80
2.8 The Northbridge (NB) . . . 81
2.8.1 Northbridge (NB) Architecture . . . 81
2.8.2 Northbridge Buffer Allocation Recommendations . . . 81
2.8.3 DMA Exclusion Vectors (DEV) . . . 81
2.8.4 Northbridge Routing . . . 81
2.8.4.1 Address Space Routing . . . 81
2.8.4.1.1 DRAM and MMIO Memory Space . . . 81
2.8.4.1.2 IO Space . . . 82
2.8.5 Physical Address Space . . . 82
2.9 DRAM Controllers (DCTs) . . . 82
2.9.1 DCT Configuration Registers . . . 84
2.9.2 DRAM Controller Direct Response Mode . . . 85
2.9.3 DCT/DRAM Initialization and Resume . . . 85
2.9.3.1 Low Voltage DDR3 . . . 86
2.9.3.2 DDR Phy Initialization . . . 86
2.9.3.2.1 Phy Voltage Level Programming . . . 87
2.9.3.2.2 DRAM Channel Frequency Change . . . 87
2.9.3.2.2.1 Requirements for DRAM Frequency Change During Training . . . 88
2.9.3.2.2.2 DCT Transmit Fifo Schedule Delay Programming . . . 88
2.9.3.2.3 Phy Fence Programming . . . 89
2.9.3.2.3.1 Phy Fence Training . . . 90
2.9.3.2.4 Phy Compensation Initialization . . . 90
2.9.3.3 SPD ROM-Based Configuration . . . 93
2.9.3.3.1 FourActWindow (Four Bank Activate Window or tFAW) . . . 94
2.9.3.4 Non-SPD ROM-Based Configuration . . . 94
2.9.3.4.1 Trdrd and TrdrdSD (Read-to-Read Timing) . . . 94
2.9.3.4.2 Twrwr and TwrwrSD (Write-to-Write Timing) . . . 95
2.9.3.4.3 Twrrd and TwrrdSD (Write-to-Read DIMM Termination Turn-around) . . . 95
2.9.3.4.4 TrwtTO (Read-to-Write Turnaround for Data, DQS Contention) . . . 95
2.9.3.4.5 DRAM ODT Control . . . 96
2.9.3.4.6 DRAM Address Timing and Output Driver Compensation Control . . . 97
2.9.3.4.7 NB P-states for DCT/Dram Initialization and Training . . . 99
2.9.3.5 DCT Training Specific Configuration . . . 100
2.9.3.6 DRAM Device Initialization . . . 101
2.9.3.6.1 Software DDR3 Device Initialization . . . 101

2.9.3.6.1.1 DDR3 MR Initialization . . . 102
2.9.3.7 DRAM Training . . . 103
2.9.3.7.1 Write Levelization Training . . . 103
2.9.3.7.1.1 Write Levelization Seed Value . . . 105
2.9.3.7.2 DQS Receiver Enable Training . . . 106
2.9.3.7.2.1 DQS Receiver Enable Training Seed Value . . . 107
2.9.3.7.3 DQS Receiver Enable Cycle Training . . . 107
2.9.3.7.4 DQS Position Training . . . 108
2.9.3.7.5 Calculating MaxRdLatency . . . 110
2.9.3.7.5.1 MaxRdLatency Training . . . 111
2.9.3.7.6 DRAM Training Pattern Generation . . . 112
2.9.3.7.6.1 Continuous Pattern Generation . . . 113
2.9.3.7.6.2 Alternative Address Mode . . . 113
2.9.3.7.6.3 Read Pattern Generation . . . 113
2.9.3.7.6.4 Write Pattern Generation . . . 114
2.9.3.8 DRAM Channel Disable . . . 114
2.9.3.9 DRAM Phy Power Savings . . . 114
2.9.4 Memory Interleaving Modes . . . 115
2.9.4.1 Chip Select Interleaving . . . 115
2.9.5 Memory Hoisting . . . 116
2.9.5.1 DctSelBaseAddr Programming . . . 117
2.9.5.2 DramHoleOffset Programming . . . 117
2.9.5.3 DctSelBaseOffset Programming . . . 118
2.9.6 DRAM CC6/PC6 Storage . . . 119
2.9.7 DRAM On DIMM Thermal Management . . . 119
2.10 Thermal Functions . . . 120
2.10.1 The Tctl Temperature Scale . . . 120
2.10.2 Sideband Temperature Sensor Interface (SB-TSI) . . . 121
2.10.3 Temperature-Driven Logic . . . 121
2.10.3.1 PROCHOT_L and Hardware Thermal Control (HTC) . . . 121
2.10.3.2 Local Hardware Thermal Control (LHTC) . . . 122
2.10.3.3 THERMTRIP . . . 123
2.11 Root complex . . . 124
2.11.1 Overview . . . 124
2.11.2 Interrupt Routing . . . 124
2.11.3 Links . . . 125
2.11.3.1 Overview . . . 125
2.11.3.2 Link Configurations . . . 125
2.11.3.3 Clocking . . . 127
2.11.4 Root Complex Configuration . . . 127
2.11.4.1 LPC MMIO Requirements . . . 127
2.11.4.2 Link Configuration and Initialization . . . 128
2.11.4.2.1 Clock Configuration . . . 128
2.11.4.2.2 Link Configuration and Core Initialization . . . 128
2.11.4.2.3 Link Training . . . 129
2.11.4.3 Miscellaneous Features . . . 131
2.11.4.3.1 Straps . . . 131
2.11.4.3.2 Lane Reversal . . . 132
2.11.4.3.3 Link Speed Changes . . . 132
2.11.4.3.3.1 Software Initiated Link Speed Changes . . . 132

2.11.4.3.3.2 Autonomous Link Speed Changes . . . 132
2.11.4.3.4 Deemphasis . . . 132
2.11.4.4 Power Management . . . 133
2.11.4.4.1 Link States . . . 133
2.11.4.4.2 Dynamic Link-width Control . . . 133
2.11.4.5 Link Test and Debug Features . . . 134
2.11.4.5.1 Compliance Mode . . . 134
2.11.5 BIOS Timer . . . 134
2.12 System Management Unit (SMU). . . . 134
2.12.1 Microcontroller . . . 134
2.12.1.1 Updating Firmware . . . 134
2.12.1.2 Software Interrupts . . . 135
2.12.2 BIOS Requirements for SMU Initialization . . . 136
2.13 Digital Display Interface (DDI) . . . 137
2.14 Graphics Processor (GPU) . . . 137
2.14.1 GPU PCI Interface . . . 137
2.14.2 Graphics Memory Controller (GMC) . . . 137
2.14.2.1 GMC Initialization . . . 137
2.14.2.1.1 Register Save/Restore Engine (RENG) . . . 138
2.14.2.1.1.1 RENG Initialization . . . 138
2.14.3 Frame Buffer (FB) . . . 139
2.15 Debug Support . . . 139
2.16 Machine Check Architecture . . . 139
2.16.1 Machine Check Registers . . . 139
2.16.2 Machine Check Errors . . . 141
2.16.2.1 Machine Check Error Logging and Reporting . . . 141
2.16.2.2 Machine Check Error Logging Overwrite During Overflow . . . 141
2.16.2.3 Machine Check Error Codes . . . 142
2.16.3 Handling Machine Check Exceptions . . . 144
2.16.3.1 ECC . . . 145
2.17 Sideband Interface (SBI). . . . 147
2.17.1 SBI Processor Information. . . . 147

**3 Registers. . . . 148**
3.1 Register Descriptions and Mnemonics . . . 148
3.1.1 Northbridge MSRs In Multi-Core Products. . . . 150
3.1.2 Software Recommendation (BIOS, SBIOS, CBIOS, etc.) . . . 150
3.2 IO Space Registers . . . 151
3.3 Device 0 Function 0 (Root Complex) Configuration Registers . . . 151
3.4 Device 1 Function 0 (Internal Graphics) Configuration Registers . . . 194
3.5 Device 1 Function 1 (Audio Controller) Configuration Registers . . . 206
3.6 Device [8:2] Function 0 (Root Port) Configuration Registers . . . 216
3.7 Device 18h Function 0 Configuration Registers. . . . 244
3.8 Device 18h Function 1 Configuration Registers. . . . 247
3.9 Device 18h Function 2 Configuration Registers. . . . 252
3.10 Device 18h Function 3 Configuration Registers. . . . 303
3.11 Device 18h Function 4 Configuration Registers. . . . 331
3.12 Device 18h Function 5 Configuration Registers. . . . 344
3.13 Device 18h Function 6 Configuration Registers. . . . 344
3.14 Device 18h Function 7 Configuration Registers. . . . 355

3.15 Internal System Management Unit (SMU) Registers . . . . . 356
3.16 Fixed Configuration Space (FCR) . . . . . 375
3.17 GPU Memory Mapped Registers . . . . . 386
3.17.1 Memory Mapped SMU Registers . . . . . 387
3.17.2 Graphics Memory Controller (GMC) Registers . . . . . 394
3.18 APIC Registers . . . . . 411
3.19 CPUID Instruction Registers . . . . . 420
3.20 MSRs - MSR0000_xxxx . . . . . 434
3.21 MSRs - MSRC000_0xxx . . . . . 453
3.22 MSRs - MSRC001_0xxx . . . . . 455
3.23 MSRs - MSRC001_1xxx . . . . . 476
3.24 Performance Counter Events . . . . . 484
3.24.1 Floating Point Events . . . . . 484
3.24.2 Load/Store and TLB Events . . . . . 486
3.24.3 Data Cache Events . . . . . 487
3.24.4 L2 Cache and System Interface Events . . . . . 490
3.24.5 Instruction Cache Events . . . . . 493
3.24.6 Execution Unit Events . . . . . 495
3.24.7 Memory Controller Events . . . . . 499
3.24.8 Crossbar Events . . . . . 503

**4 Register List . . . . . 505**

# List of Figures

Figure 1: A processor ..... 22
Figure 2: System Diagram ..... 23
Figure 3: C-state decision process ..... 55
Figure 4: NBP0 to NBP1 transition determination ..... 64
Figure 5: NBP1 to NBP0 transition determination ..... 65
Figure 6: DQS Position Training Example Results ..... 109
Figure 7: DQS Position Training Insertion Delay Recovery Example Results ..... 110
Figure 8: Example cases for programming DramHoleOffset ..... 118
Figure 9: Example cases for programming DctSelBaseOffset ..... 119
Figure 10: Tctl scale ..... 121
Figure 11: Root complex topology ..... 124
Figure 12: Phy clock configuration ..... 127
Figure 13: Phy recovered clock and sample clock ..... 178
Figure 14: Address/command timing at the processor pins ..... 268

# List of Tables

Table 1: Definitions ... 17
Table 2: SMM initial state ... 33
Table 3: SMM save state ... 33
Table 4: Power management support ... 40
Table 5: Software P-state numbering example ... 47
Table 6: P-state control example ... 48
Table 7: PSPP Maximum PCIe Speed ... 76
Table 8. Recommended buffer settings ... 81
Table 9: DCT Definitions ... 83
Table 10: DDR3 UDIMM Maximum Frequency Support for FM1 ... 83
Table 11: DDR3 SO-DIMM Maximum Frequency Support for FM1 ... 83
Table 12: DDR3 SO-DIMM Maximum Frequency Support for FS1 ... 84
Table 13: DDR3 UDIMM and SO-DIMM Population Support for FM1 ... 84
Table 14: DDR3 SO-DIMM Population Support for FS1 ... 84
Table 15: Phy predriver calibration codes for Data/DQS at 1.5V ... 90
Table 16: Phy predriver calibration codes for Data/DQS at 1.35V ... 91
Table 17: Phy predriver calibration codes for Cmd/Addr at 1.5V ... 91
Table 18: Phy predriver calibration codes for Cmd/Addr at 1.35V ... 92
Table 19: Phy predriver calibration codes for Clock at 1.5V ... 92
Table 20: Phy predriver calibration codes for Clock at 1.35V ... 93
Table 21: DIMM ODT settings ... 96
Table 22: DIMM ODT pattern ... 97
Table 23: BIOS recommendations for SO-DIMM address timings and output driver control ... 97
Table 24: BIOS recommendations for UDIMM address timings and output driver control ... 98
Table 25: DCT training specific register values ... 100
Table 26. DDR3 MR0 ... 102
Table 27. DDR3 MR1 ... 102
Table 28. DDR3 MR2 ... 103
Table 29. DDR3 MR3 ... 103
Table 30. Recommended interleave configurations ... 115
Table 31. DDR3 swapped normalized address lines for CS interleaving ... 116
Table 32. C6Base Programming Example ... 119
Table 33: INTx Mapping ... 125
Table 34: Lane Id Mapping ... 125
Table 35: Supported Gfx Port Configurations ... 125
Table 36: Supported General Purpose (GPP) Link Configurations ... 126
Table 37: Microcontroller Interrupt Service Indices ... 135
Table 38: Supported DDI Modes ... 137
Table 39: RENG Initialization Sequence ... 138
Table 40: Recommended Frame Buffer Configurations ... 139

Table 41: MCA register cross-reference table ... 140
Table 42: MC0 and MC4 overwrite priorities ... 142
Table 43: MC1, MC2, MC3, and MC5 overwrite priorities ... 142
Table 44: Error code types ... 143
Table 45: Error codes: transaction type ... 143
Table 46: Error codes: cache level ... 143
Table 47: Error codes: memory transaction type ... 143
Table 48: Error codes: participation processor ... 144
Table 49: Error codes: memory or IO ... 144
Table 50: ECC correctable syndromes ... 145
Table 51: Terminology in register descriptions ... 149
Table 52: Index address mapping for D0F0x64_x5[B,9,7,5,3,1] ... 159
Table 53: Index addresses for D0F0xE4_x0[2:1]1[2:0]_0010 ... 169
Table 54: Index addresses for D0F0xE4_x0[2:1]1[2:0]_0011 ... 170
Table 55: Index addresses for D0F0xE4_x0[2:1]1[2:0]_001[3:2] ... 171
Table 56: Index addresses for D0F0xE4_x0[2:1]1[2:0]_0015 ... 172
Table 57: Per phy register addresses to pin mappings ... 172
Table 58: Per nibble register addresses to pin mappings ... 172
Table 59: Phy per receiver lane register addresses ... 178
Table 60: Phy receiver broadcast register addresses ... 178
Table 61: Recommended DCV settings ... 181
Table 62: Phy per transmitter lane register addresses ... 182
Table 63: Phy transmitter broadcast register addresses ... 183
Table 64: Recommended link configuration ... 183
Table 65: Recommended preemphasis settings ... 184
Table 66: Index address mapping for D0F0xE4_x013[1:0]_0[C:8]00 ... 185
Table 67: Index address mapping for D0F0xE4_x013[1:0]_0[C:8]03 ... 185
Table 68: Reserved field mappings for D0F0xE4_x013[2:0]_8013 ... 187
Table 69: Reserved field mappings for D0F0xE4_x013[2:0]_8014 ... 188
Table 70: Reserved field mappings for D0F0xE4_x013[2:1]_8040 ... 194
Table 71: Reset mapping for D[8:2]F0x00. ... 216
Table 72: Link controller state encodings ... 236
Table 73: DIMM, Chip Select, CKE, ODT, and Register Mapping ... 253
Table 74: DDR3 DRAM address mapping ... 258
Table 75: Index addresses for D18F2x[1,0]9C_x0000_0[1:0]0[2:1] ... 267
Table 76: Byte lane mapping for D18F2x[1,0]9C_x0000_0[1:0]0[2:1] ... 267
Table 77: Index addresses for D18F2x[1,0]9C_x0000_0[1:0]0[6:5] ... 269
Table 78: Byte lane mapping for D18F2x[1,0]9C_x0000_0[1:0]0[6:5] ... 269
Table 79. DRAM Phy PLL Multiplier and Divide Ratio Values ... 269
Table 80: Index addresses for D18F2x[1,0]9C_x0000_00[24:10] ... 273
Table 81: Byte lane mapping for D18F2x[1,0]9C_x0000_00[24:10] ... 273
Table 82: Index addresses for D18F2x[1,0]9C_x0000_00[44:30] ... 274
Table 83: Byte lane mapping for D18F2x[1,0]9C_x0000_00[44:30] ... 274

Table 84: Index addresses for D18F2x[1,0]9C_x0000_00[51:50] ..... 275
Table 85: Byte lane mapping for D18F2x[1,0]9C_x0000_00[51:50] ..... 275
Table 86: Index addresses for D18F2x[1,0]9C_x0D0F_0[F,7:0]02 ..... 276
Table 87: Broadcast write index address for D18F2x[1,0]9C_x0D0F_0[F,7:0]02 ..... 276
Table 88: Valid values for D18F2x[1,0]9C_x0D0F_0[F,7:0]02[TxPreP, TxPreN] ..... 276
Table 89: Index addresses for D18F2x[1,0]9C_x0D0F_0[F,7:0]0[A,6] pad group 1 ..... 277
Table 90: Index addresses for D18F2x[1,0]9C_x0D0F_0[F,7:0]0[A,6] pad group 2 ..... 277
Table 91: Broadcast write index address for D18F2x[1,0]9C_x0D0F_0[F,7:0]0[A,6] ..... 277
Table 92: Index addresses for D18F2x[1,0]9C_x0D0F_0[F,7:0]0F ..... 277
Table 93: Broadcast write index address for D18F2x[1,0]9C_x0D0F_0[F,7:0]0F ..... 277
Table 94: Index addresses for D18F2x[1,0]9C_x0D0F_0[F,7:0]10 ..... 278
Table 95: Broadcast write index address for D18F2x[1,0]9C_x0D0F_0[F,7:0]10 ..... 278
Table 96: Index addresses for D18F2x[1,0]9C_x0D0F_0[F,7:0]13 ..... 278
Table 97: Broadcast write index address for D18F2x[1,0]9C_x0D0F_0[F,7:0]13 ..... 279
Table 98: Index addresses for D18F2x[1,0]9C_x0D0F_0[F,7:0]1F ..... 279
Table 99: Broadcast write index address for D18F2x[1,0]9C_x0D0F_0[F,7:0]1F ..... 279
Table 100: Index addresses for D18F2x[1,0]9C_x0D0F_0[F,7:0]30 ..... 280
Table 101: Broadcast write index address for D18F2x[1,0]9C_x0D0F_0[F,7:0]30 ..... 280
Table 102: Index addresses for D18F2x[1,0]9C_x0D0F_0[F,7:0]31 ..... 280
Table 103: Broadcast write index address for D18F2x[1,0]9C_x0D0F_0[F,7:0]31 ..... 281
Table 104: Index address mapping for D18F2x[1,0]9C_x0D0F_2[1:0]02 ..... 281
Table 105: Index address mapping for D18F2x[1,0]9C_x0D0F_[C,8,2][F,1:0]1F ..... 282
Table 106: Broadcast write index address for D18F2x[1,0]9C_x0D0F_[C,8,2][F,1:0]1F ..... 282
Table 107: Index address mapping for D18F2x[1,0]9C_x0D0F_2[1:0]20 ..... 282
Table 108: Index address mapping for D18F2x[1,0]9C_x0D0F_2[1:0]30 ..... 283
Table 109: MemClkDis mapping for D18F2x[1,0]9C_x0D0F_2[1:0]30 ..... 283
Table 110: Index address mapping for D18F2x[1,0]9C_x0D0F_[C,8,2][F,1:0]31 ..... 283
Table 111: Broadcast write index address for D18F2x[1,0]9C_x0D0F_[C,8,2][F,1:0]31 ..... 284
Table 112: Index address mapping for D18F2x[1,0]9C_x0D0F_[C,8][1:0]02 ..... 284
Table 113: Index address mapping for D18F2x[1,0]9C_x0D0F_[C,8][1:0][12,0E,0A,06] ..... 285
Table 114: BIOS Recommendations for D18F2x[1,0]F4_x200[Txp] ..... 294
Table 115: BIOS Recommendations for D18F2x[1,0]F4_x200[Txpdll] ..... 294
Table 116: Address mapping for D18F2x[1E0:1D8] ..... 302
Table 117: NB error descriptions ..... 308
Table 118: NB error signatures, part 1 ..... 308
Table 119: NB error signatures, part 2 ..... 309
Table 120: SMAF to ACPI state mapping ..... 314
Table 121: Internal unit ID mapping ..... 323
Table 122: C-state action field definition ..... 333
Table 123: LCLK DPM Sampling Period [4:0] to DPM State Mapping ..... 359
Table 124: LCLK PCIe Up Hysteresis [2:0] to DPM State Mapping ..... 359
Table 125: LCLK DPM Scaling Divisor [1:0] to DPM State Mapping ..... 360
Table 126: LCLK DPM Scaling VID [2:0] to DPM State Mapping ..... 361

Table 127: LCLK PCIe Down Hysteresis [2:0] to DPM State Mapping ..... 363
Table 128: LCLK PCIe Scaling Divisor [2:0] to DPM State Mapping ..... 363
Table 129: LCLK PCIe Scaling Divisor [2:0] to DPM State Mapping ..... 364
Table 130: PCIe Lane Count Thresholds [4:0] to PCIe Bandwidth State Mapping ..... 373
Table 131: SCLK DPM Control [1:0] to DPM State Mapping ..... 388
Table 132: SclkDpmGnbSlow to SCLK DPM State Mapping ..... 390
Table 133: SCLK deep sleep to DPM state mapping ..... 391
Table 134: DPM state mapping for GMMx7[6C:50] ..... 392
Table 135: SCLK shallow sleep to DPM state mapping ..... 393
Table 136: GMC CS base address register mapping and reset values ..... 404
Table 137: GMC CS mask register mapping ..... 404
Table 138: GMC bank address register mapping ..... 405
Table 139: Div[3,1:0] Value Table ..... 418
Table 140: Reset mapping for CPUID Fn0000_0000_E[B,C,D]X. ..... 421
Table 141: Reset mapping for CPUID Fn8000_0000_EAX. ..... 425
Table 142: Reset mapping for CPUID Fn8000_000[4,3,2]_E[D,C,B,A]X. ..... 427
Table 143: Fixed-size MTRR size and range mapping ..... 439
Table 144: DC error descriptions ..... 444
Table 145: DC error signatures ..... 444
Table 146: DC error data; address register ..... 445
Table 147: IC error descriptions ..... 446
Table 148: IC error signatures ..... 447
Table 149: IC error data; address register ..... 448
Table 150: BU error descriptions ..... 449
Table 151: BU error signatures ..... 449
Table 152: BU error data; address register ..... 450
Table 153: LS error signatures ..... 451
Table 154: FR error signatures ..... 453

# Revision History

**Revision 3.02 Changes**

- Updated 2.9 [DRAM Controllers (DCTs)].
- Updated 2.5.3.1.8.
- Updated 2.11.4.5.1.
- Updated D18F2x[1,0]9C_x0D0F_4009.
- Updated D18F3xDC[HwPstateMaxVal].
- Updated D[8:2]F0xA0, D1F1xA0.
- Updated SMUx0B_x85B0.
- Updated CPUID Fn8000_0001_EBX.
- Updated CPUID Fn8000_0007_EDX.
- Updated MSRC001_0015[INVDWBINVD].
- Updated MSRC001_1021[DIS_SPEC_TLB_RLD].
- Updated MSRC001_1022[DIS_SPEC_TLB_RLD].
- Updated MSRC001_1022[DIS_HW_PF].
- Added MSRC001_0055.
- Updated MSRC001_00[6B:64].
- Added MSRC001_1036[NbIbsReqCacheHitSt, NbIbsReqDstNode].

**Revision 3.00 Changes**

- Initial Public Release.

# 1 Overview

The AMD family 12h processor (in this document referred to as *the processor*) is a processing unit that supports x86-based instruction sets. The processor includes (a) one to four independent central processing unit cores (referred to as *cores*), (b) one PCIe® root complex (in this document referred to as the root complex or RC) with generation 2 link support, (c) up to two DDR3 system memory DRAM interfaces, and optionally (d) a graphics core.

AMD family 12h processors are distinguished by the combined ExtFamily and BaseFamily fields of the CPUID instruction (see CPUID Fn0000_0001_EAX).

## 1.1 Intended Audience

This document provides the processor behavioral definition and associated design notes. It is intended for platform designers and for programmers involved in the development of low-level BIOS (basic input/output system) functions, drivers, and operating system kernel modules. It assumes prior experience in personal computer platform design, microprocessor programming, and legacy x86 and AMD64 microprocessor architecture. The reader should also have familiarity with various platform technologies, such as DDR DRAM.

## 1.2 Reference Documents

- Advanced Configuration and Power Interface (ACPI) Specification. www.acpi.info.
- AMD64 Architecture Programmer's Manual Volume 1: Application Programming, #24592.
- AMD64 Architecture Programmer's Manual Volume 2: System Programming, #24593.
- AMD64 Architecture Programmer's Manual Volume 3: Instruction-Set Reference, #24594.
- AMD64 Architecture Programmer's Manual Volume 4: 128-Bit Media Instructions, #26568.
- AMD64 Architecture Programmer's Manual Volume 5: 64-Bit Media and x87 Floating-Point Instructions, #26569.
- AMD Family 12h Processor Electrical Data Sheet, #41609.
- AMD Voltage Regulator Specification, #40182.
- CPUID Specification, #25481.
- PCI local bus specification. (www.pcisig.org).
- PCI Express specification. (www.pcisig.org).
- System Management Bus (SMBus) specification. (www.smbus.org).

## 1.3 Conventions

### 1.3.1 Numbering

- **Binary numbers**. Binary numbers are indicated by appending a “b” at the end, e.g., 0110b.
- **Decimal numbers**. Unless specified otherwise, all numbers are decimal. This rule does not apply to the register mnemonics described in see 3.1 [Register Descriptions and Mnemonics].
- **Hexadecimal numbers**. Hexadecimal numbers are indicated by appending an “h” to the end, e.g., 45f8h.
- **Underscores in numbers**. Underscores are used to break up numbers to make them more readable. They do not imply any operation. E.g., 0110_1100b.

### 1.3.2 Arithmetic And Logical Operators

In this document, formulas follow some Verilog conventions for logic equations.

| | |
|---|---|
| {} | Curly brackets are used to indicate a group of bits that are concatenated together. Each set of bits is separated by a comma. E.g., {Addr[3:2], Xlate[3:0]} represents a 6-bit value; the two MSBs are Addr[3:2] and the four LSBs are Xlate[3:0]. |
| \| | Logical OR operator. |
| & | Logical AND operator. |
| ^ | Logical exclusive-OR operator; sometimes used as “raised to the power of” as well, as indicated by the context in which it is used. |
| ~ | Logical NOT operator. |
| == | Logical “is equal to” operator. |
| != | Logical “is not equal to” operator. |
| <= | Less than or equal operator. |
| >= | Greater than or equal operator. |
| * | Arithmetic multiplied-by operator. |
| ABS | Absolute value. |
| CEIL | Round up to the next integer value. |
| FLOOR | Round down to the next integer value. |
| MAX | Picks maximum integer or real value of comma separated list. |
| ROUND | Round to the nearest integer. |

The order in which logical operators are applied is: ~ first, & second, and | last.

For example, the equation:

```
Output[3:0] = {A[1:0], B[3:2]} & C[3:0] | ~D[3:0] & E[9:6],
```

is translated as:

```
Output[3] = (A[1] & C[3]) | (~D[3] & E[9]);
Output[2] = (A[0] & C[2]) | (~D[2] & E[8]);
Output[1] = (B[3] & C[1]) | (~D[1] & E[7]);
Output[0] = (B[2] & C[0]) | (~D[0] & E[6]);
```

## 1.4 Definitions

**Table 1: Definitions**

| Term | Definition |
|---|---|
| **AP** | Application processor. See 2.3 [Processor Initialization]. |
| **BatteryPower** | The system is running from a battery power source or otherwise undocked from a continuous power supply. Settings using this definition may be required to change during runtime. |
| **BCS** | Base configuration space. See 2.7 [Configuration Space]. |
| **BERT** | Bit error rate tester. A piece of test equipment that generates arbitrary test patterns and checks that a device under test returns them without errors. |
| **BIST** | Built-in self-test. Hardware within the processor that generates test patterns and verifies that they are stored correctly (in the case of memories) or received without error (in the case of links). |
| **Boosted P-state** | Any P-state with higher performance than software P0. See 2.5.3.1.1 [Core Performance Boost (CPB)] and 2.5.3.1.2 [Core P-state Naming and Numbering]. |

**Table 1: Definitions**

| Term | Definition |
|---|---|
| **BoostEn** | BoostEn = ((D18F4x15C[NumBoostStates] != 0) && (D18F4x15C[BoostSrc] != 0)). |
| **Boot VID** | Boot voltage ID. This is the VDD and VDDNB voltage level that the processor requests from the external voltage regulator during the initial phase of the cold boot sequence. |
| **BSP** | Boot strap processor. See 2.3 [Processor Initialization]. |
| **C-states** | These are ACPI-defined CPU power states. C0 is operational. All other C-states are low-power states in which the processor is not executing code. See 2.5.3.2 [C-states]. |
| **CAF** | C-state action field. See D18F4x118 and D18F4x11C. |
| **Canonical address** | An address in which the state of the most-significant implemented bit is duplicated in all the remaining higher-order bits, up to bit 63. |
| **Channel** | See DRAM channel. |
| **Channel interleaved mode** | Mode in which DRAM address space is interleaved between DRAM channels. See 2.9.4 [Memory Interleaving Modes]. |
| **CMP** | Chip multi-processing. Refers to processors that include multiple cores. See 2.1 [Processor Overview]. |
| **NBCIF** | NB to core interface. |
| **Coherent fabric** | The coherent fabric includes the DRAM controllers and caches of the system. See 2.2 [System Overview]. |
| **COF** | Current operating frequency of a given clock domain. See 2.5.3.1 [Core P-states]. |
| **Cold reset** | PWROK is deasserted and RESET_L is asserted. See 2.3 [Processor Initialization]. |
| **CPU or core** | The instruction execution unit of the processor. See 2.1 [Processor Overview]. |
| **CpuCoreNum** | CpuCoreNum=CPUID Fn0000_0001_EBX[LocalApicId][1:0]. Specifies the core number. See 2.4.2 [Processor Cores and Downcoring]. |
| **CPUID function X** | Refers to the CPUID instruction when EAX is preloaded with X. See 3.19 [CPUID Instruction Registers]. |
| **CS** | Chip select. See D18F2x[1,0][4C:40] [DRAM CS Base Address Registers]. |
| **DCT** | DRAM controller. See 2.9 [DRAM Controllers (DCTs)]. |
| **DEV** | DMA exclusion vector. See 2.8.3 [DMA Exclusion Vectors (DEV)]. |
| **DID** | Divisor identifier. Specifies the post-PLL divisor used to reduce the COF. See 2.5.3.1 [Core P-states]. |
| **Display refresh** | Traffic used for display refresh in UMA systems. |
| **DMA** | Direct memory access. An access to memory that does not use the CPU. |
| **Doubleword** | A 32-bit value. |
| **Downcoring** | Removal of cores. See 2.4.2 [Processor Cores and Downcoring]. |
| **DRAM channel** | The part of the DRAM interface that connects to a 64-bit DIMM. For example, a processor with a 128-bit DRAM interface is said to support two DRAM channels. See 2.9 [DRAM Controllers (DCTs)]. |
| **DS** | Downstream. Refers to the direction of data on a link. In the context of a memory buffer link, this refers to data flow from the controller to the memory buffer. |
| **DualChannel** | DualChannel = (D18F2x094[DisDramInterface]==0 && D18F2x194[DisDramInterface]==0). |
| **Dual-Plane** | Refers to a processor or systemboard where VDD and VDDNB are separate and may operate at independent voltage levels. See 2.5.1 [Processor Power Planes And Voltage Control]. |
| **DW or DWORD** | Doubleword. A 32-bit value. |

**Table 1: Definitions**

| Term | Definition |
|---|---|
| **ECS** | Extended configuration space. See 2.7 [Configuration Space]. |
| **EDS** | Electrical data sheet. See 1.2 [Reference Documents]. |
| **FCH** | Fusion controller hub. The platform device that contains the bridge to the system BIOS. |
| **FDS** | Functional data sheet; there is one FDS for each package type. |
| **FID** | Frequency identifier. Specifies the PLL frequency multiplier for a given clock domain. See 2.5.3.1 [Core P-states]. |
| **Ganged** | A link, memory channel, or voltage regulator in which all portions are controlled as one. |
| **GB or Gbyte** | Gbyte or Gigabyte; 1,073,741,824 bytes. |
| **#GP** | A general-protection exception. |
| **#GP(0)** | Notation indicating a general-protection exception (#GP) with error code of 0. |
| **GpuEnabled** | GpuEnabled = ( D1F0x00 != FFFF_FFFFh ). |
| **GT/s** | Giga-transfers per second. |
| **HTC** | Hardware thermal control. See 2.10.3.1 [PROCHOT_L and Hardware Thermal Control (HTC)]. |
| **HTC-active state** | Hardware-controlled lower-power, lower-performance state used to reduce temperature. See 2.10.3.1 [PROCHOT_L and Hardware Thermal Control (HTC)]. |
| **IBS** | Instruction based sampling. See 2.6 [Performance Monitoring]. |
| **IFQ** | In flight queue. |
| **IO configuration** | Access to configuration space through IO ports CF8h and CFCh. See 2.7 [Configuration Space]. |
| **IORR** | IO range register. See [The IO Range Registers Base (IORR_BASE[1:0])] MSRC001_00[18,16]. |
| **KB or Kbyte** | Kilobyte; 1024 bytes. |
| **L1 caches** | The level 1 caches of the core including the instruction cache and the data cache. |
| **L2 cache** | The level 2 cache of each core. |
| **Linear (virtual) address** | The address generated by a core after the segment is applied. |
| **Link** | Generic term that refers to a PCIe® link. |
| **LINT** | Local interrupt. |
| **Logical address** | The address generated by a core before the segment is applied. |
| **LVT** | Local vector table. A collection of APIC registers that define interrupts for local events. E.g., [The Extended Interrupt [3:0] Local Vector Table Registers] APIC[530:500]. |
| **Master abort** | This is a PCI-defined term that is applied to transactions on other than PCI buses. It indicates that the transaction is terminated without affecting the intended target; reads return all 1's; writes are discarded; the master abort error code is returned in the response, if applicable; master abort error bits are set if applicable. |
| **MB or Mbyte** | Megabyte; 1024 Kilobytes. |
| **MEMCLK** | Refers to the clock signals, M[B, A][3:0]_CLK, that are driven from the processor to DDR DIMMs. The MEMCLK frequency is determined by D18F2x[1,0]94[MemClkFreq]. |

**Table 1: Definitions**

| Term | Definition |
|---|---|
| **Micro-op** | Micro-op. Instructions have variable-length encoding and many perform multiple primitive operations. The processor does not execute these complex instructions directly, but, instead, decodes them internally into simpler fixed-length instructions called macro-ops. The processor scheduler breaks down macro-ops into sequences of even simpler instructions called micro-ops, each of which specifies a single primitive operation. |
| **MMIO** | Memory-mapped input-output range. This is physical address space that is mapped to the IO functions such as the links or MMIO configuration. The link MMIO ranges are specified by [The Memory Mapped IO Base Registers] D18F1x[B8,B0,A8,A0,98,90,88,80]. |
| **MMIO configuration** | Access to configuration space through memory space. See 2.7 [Configuration Space]. |
| **MODXXSY or MODXX** | Specifies the processor model and stepping. XX is the hexadecimal model number. Y is the hexadecimal stepping. |
| **MOF** | Maximum operating frequency of the core(s). Normally this is the core COF in P-state 0. See 2.5.3.1 [Core P-states]. |
| **MSR** | Model specific register. The CPU includes several MSRs for general configuration and control. See 3.20 [MSRs - MSR0000_xxxx] for the beginning of the MSR register definitions. |
| **MTRR** | Memory-type range register. The MTRRs specify the type of memory associated with various memory ranges. See MSR0000_00FE, MSR0000_020[E,C,A,8,6,4,2,0], MSR0000_02[6F:68,59,58,50], and MSR0000_02FF. |
| **NB** | Northbridge. The transaction routing block of the node. See 2.1 [Processor Overview]. |
| **NCLK** | The main northbridge clock. The NCLK frequency is the NB COF. |
| **Node** | See 2.1 [Processor Overview]. |
| **Normalized address** | Addresses used by DCTs. See 2.9 [DRAM Controllers (DCTs)]. |
| **Octword** | A 128-bit value. |
| **ODM** | On-DIMM mirroring. See D18F2x[1,0][4C:40][OnDimmMirror]. |
| **ODT** | On-die termination, which is applied DRAM interface signals. |
| **ODTS** | DRAM On-die thermal sensor. |
| **Operational frequency** | The frequency at which the processor operates. See 2.5 [Power Management]. |
| **PCIe®** | PCI Express. |
| **PDS** | Product data sheet. |
| **Physical address** | Addresses used by cores in transactions sent to the NB. |
| **PRBS** | Pseudo-random bit sequence. |
| **Processor** | See 2.1 [Processor Overview]. |
| **P-state** | Performance state. See 2.5 [Power Management]. |
| **PTE** | Page table entry. |
| **PVI** | Parallel VID interface. See 2.5.1 [Processor Power Planes And Voltage Control]. |
| **Quadword** | A 64-bit value. |
| **READ** | Used in an IF statement to define a register address that has a unique definition for read and write accesses. |
| **Root complex or RC** | See the PCI Express® 2.0 Base Specification. |

**Table 1: Definitions**

| Term | Definition |
|---|---|
| **RX** | Receiver. |
| **SBI** | Sideband Interface. See 2.17 [Sideband Interface (SBI)]. |
| **Shutdown** | A state in which the affected core waits for either INIT, RESET, or NMI. When shut-down state is entered, a shutdown special cycle is sent on the IO links. |
| **Slam** | Refers to change the voltage to a new value in one step (as opposed to stepping). See 2.5.1.5.1 [Hardware-Initiated Voltage Transitions]. |
| **SMAF** | System management action field. This is the code passed from the SMC to the processors in STPCLK assertion messages. The action taken by the processors in response to this message is specified by D18F3x80 [ACPI Power State Control Low]. |
| **SMBus** | System management bus. Refers to the protocol on which the serial VID interface (SVI) commands and SBI are based. See 2.5.1 [Processor Power Planes And Voltage Control], 2.17 [Sideband Interface (SBI)], and 1.2 [Reference Documents]. |
| **SMC** | System management controller. This is the platform device that communicates system management state information to the processor through an IO link, typically the FCH. |
| **SMI** | System management interrupt. See 2.4.6.2.1 [SMM Overview]. |
| **SMM** | System management mode. See 2.4.6.2 [System Management Mode (SMM)]. |
| **SmmEntry** | Used in an IF statement to define a field that has a unique definition during entry and exit of system management mode. See SMMFEC9 [Auto Halt Restart Offset]. |
| **Speculative event** | A performance monitor event counter that counts all occurrences of the event even if the event occurs during speculative code execution. |
| **STC** | Software thermal control. |
| **SVI** | Serial VID interface. See 2.5.1 [Processor Power Planes And Voltage Control]. |
| **SVM** | Secure virtual machine. See 2.4.7 [Secure Virtual Machine Mode (SVM)]. |
| **TCC** | Temperature calculation circuit. See 2.10 [Thermal Functions]. |
| **TDP** | Thermal design power. A power consumption parameter that is used in conjunction with thermal specifications to design appropriate cooling solutions for the processor. |
| **TSC** | Time stamp counter. |
| **TX** | Transmitter. |
| **UI** | Unit interval. This is the amount of time equal to one half of a clock cycle. |
| **UMI** | Unified Media Interface. The link between the processor and the FCH. |
| **US** | Upstream. Refers to the direction of data on a link. |
| **usec or us** | Microsecond. |
| **VDD** | Main power supply to the processor core logic. |
| **VDDNB** | Main power supply to the processor NB logic. |
| **VID** | Voltage level identifier. See 2.5.1 [Processor Power Planes And Voltage Control]. |
| **Virtual CAS** | The clock in which CAS is asserted for the burst, N, plus the burst length (in MEM-CLKs), minus 1; so the last clock of virtual CAS = N + (BL/2) - 1. |
| **VRM** | Voltage regulator module. |
| **Warm reset** | RESET_L is asserted only (while PWROK stays high). See 2.3 [Processor Initialization]. |
| **WDT** | Watchdog timer. A timer that detects activity and triggers an error if a specified period of time expires without the activity. For example, see [The CPU Watchdog Timer Register (CpuWdtCfg)] MSRC001_0074 or the NB watchdog timer in D18F3x40. |
| **XBAR** | Cross bar; command packet switch. See 2.8.1 [Northbridge (NB) Architecture]. |

# 2 Functional Description

## 2.1 Processor Overview

The *processor* is a package that contains (1) one to four cores, (2) one PCIe® root complex with generation 2 link support, (3) two 64-bit DDR3 interfaces for communication to system memory, and (4) one communication packet routing block referred to as the *northbridge* (NB).

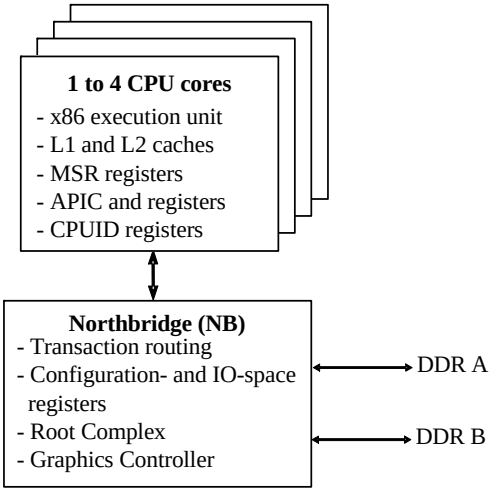


**Figure 1: A processor**

Each *core* includes x86 instruction execution logic, a first-level (L1) data cache, a first-level instruction cache, and a second level (L2) general-purpose cache. There is a set of model-specific registers (MSRs) and APIC registers associated with each core. Processors that include multiple cores are said to incorporate *chip multiprocessing* or CMP.

The links are input-output links, as defined by the PCI Express Base Specification.

Each DRAM interface supports a 64-bit DDR3 unbuffered DIMM channel.

The NB routes transactions between the cores, the links, and the DRAM interfaces. It includes the configuration register space for the device.

## 2.2 System Overview

The following diagram illustrates the expected system architecture.

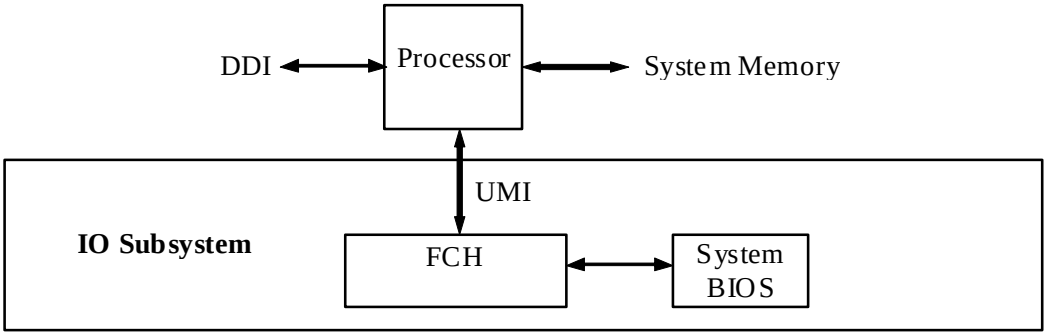


**Figure 2: System Diagram**

## 2.3 Processor Initialization

This section describes the initialization sequence after a cold reset.

Core 0 of the processor begins executing code from the reset vector. The remaining cores do not fetch code until their enable bits are set ([The Link Transaction Control Register] D18F0x68 [Cpu1En] and [The Extended Link Transaction Control Register] D18F0x168[Cpu2En,Cpu3En]).

### 2.3.1 BSP initialization

The BSP must perform the following tasks as part of the boot process.

- Store BIST information from the eax register into an unused processor register.
- If supported, determine the type of startup from either the keyboard controller or the [The Link Initialization Control Register] D18F0x6C[InitDet] bit. If this boot sequence was caused by a an INIT then BIOS vectors away from the cold/warm reset initialization path.
- Determine the history of this reset using the [The Link Initialization Control Register] D18F0x6C [ColdRstDet] bit. If this is a cold reset then BIOS must clear the [MCi_STATUS] MSRs (MSR0000_0401, MSR0000_0405, MSR0000_0409, MSR0000_040D, MSR0000_0411, MSR0000_0415). If this is a warm reset then BIOS may check for valid MCA errors and if present save the status for use later (see 2.16.3 [Handling Machine Check Exceptions]).
- Enable the cache, program the MTRRs for Cache-as-RAM and initialize the Cache-as-RAM, as described in 2.3.3 [Cache Initialization For General Storage During Boot].
- Setup the SMU (2.12.2 [BIOS Requirements for SMU Initialization]).
- Setup the local APIC (2.4.6.1.1 [ApicId Enumeration Requirements]).
- Setup the link configuration (2.11.3.2 [Link Configurations]).
- Setup the root complex and initialize the I/O links (2.11.4.2 [Link Configuration and Initialization]).
- Configure the DRAM controllers (2.9.3 [DCT/DRAM Initialization and Resume]).
- Device enumeration for all I/O-link devices (see link specification).
- Configure processor power management (see 2.5 [Power Management]).
- If supported, allow other cores to beginning fetching instructions by setting D18F0x68[Cpu1En] and D18F0x168[Cpu2En,Cpu3En].

### 2.3.2 AP initialization

All other processor cores other than core 0 begin executing code from the reset vector. They must perform the

following tasks as part of the boot process.

- Store BIST information from the eax register into an unused processor register.
- If supported, determine the type of startup from either the keyboard controller or the [The Link Initialization Control Register] D18F0x6C[InitDet] bit. If this boot sequence was caused by a an INIT then BIOS vectors away from the cold/warm reset initialization path.
- Determine the history of this reset using the [The Link Initialization Control Register] D18F0x6C [ColdRstDet] bit. If this is a cold reset then BIOS must clear the [MCi_STATUS] MSRs (MSR0000_0401, MSR0000_0405, MSR0000_0409, MSR0000_040D, MSR0000_0411, MSR0000_0415). If this is a warm reset then BIOS may check for valid MCA errors and if present save the status for use later (see 2.16.3 [Handling Machine Check Exceptions]).
- Setup the local APIC (2.4.6.1.1 [ApicId Enumeration Requirements]).
- Configure processor power management (see 2.5 [Power Management]).

### 2.3.3 Cache Initialization For General Storage During Boot

Prior to initializing the DRAM controller for system memory, BIOS may use the L2 cache of each core as general storage.

The L2 cache as storage is described as follows:
- Each core has its own L2 cache.
- BIOS manages the mapping of the L2 storage such that cacheable accesses do not cause L2 victims.
- The L2 size, L2 associativity, and L2 line size is determined by reading CPUID Fn8000_0006_ECX[L2Size, L2Assoc, L2LineSize]. L2WayNum is defined to be the number of ways indicated by the L2Assoc code.
  - The L2 cache is viewed as (L2Size/L2LineSize) cache lines of storage, organized as L2WayNum ways, each way being (L2Size/L2WayNum) in size.
    - E.g. L2Assoc=8 so L2WayNum=16 (there are 16 ways). If L2Size=512KB then there are 16 blocks of cache, each 512KB/16 in size, or 32KB each.
  - For each of the following values of L2Size, the following values are defined:
    - L2Size=512KB: L2Tag=PhysAddr[39:16], L2WayIndex=PhysAddr[15:6].
    - L2Size=1MB: L2Tag=PhysAddr[39:17], L2WayIndex=PhysAddr[16:6].
  - PhysAddr[5:0] addresses the L2LineSize number of bytes of storage associated with the cache line.
  - The L2 cache, when allocating a line at L2WayIndex, will:
    - Pick an invalid way before picking a valid way.
    - Prioritize the picking of invalid ways such that way 0 is the highest priority and L2WayNum-1 is the lowest priority.
  - It is recommended that BIOS assume a simpler allocation of L2 cache memory, being L2WayNum size-aligned blocks of memory, each being L2Size/L2WayNum bytes.
  - BIOS can rely on a minimum L2Size of 512 KB for Fam12h. See CPUID Fn8000_0006_ECX[L2Size].

- In order to prevent victimizing L2 data, no more than L2WayNum cache lines may have the same L2WayIndex.
  - Software does not need to know which ways the L2WayNum lines are allocated to for any given value of L2WayIndex, only that invalid ways will be selected for allocation before valid ways will be selected for allocation.

The following memory types are supported as follows:
  - WP-IO: BIOS ROM may be assigned the write-protect IO memory type and may be accessed read-only as data and fetched as instructions.
    - BIOS initializes a location in the L2 cache, mapped as write-protect IO, with 1 load of any size or an instruction fetch to any location within the L2LineSize cache line.

  - WB-DRAM: General storage may be assigned the write-back DRAM memory type and may be accessed as read-write data, but not accessed by instruction fetch.
    - BIOS initializes an L2LineSize sized and aligned location in the L2 cache, mapped as write-back DRAM, with 1 read to at least 1 byte of the L2LineSize sized and aligned WB-DRAM address. BIOS may store to a line only after it has been allocated by a load.
    - Fills, sent to the disabled memory controller, return undefined data.
  - All of memory space that is not accessed as WP-IO or WB-DRAM space must be marked as UC memory type.

- In order to prevent victimizing L2 data, no more than L2WayNum cache lines accessed as WP-IO or WB-DRAM may have the same L2WayIndex.
  - Software does not need to know which ways the L2WayNum lines are allocated to for any given value of L2WayIndex, only that invalid ways will be selected for allocation before valid ways will be selected for allocation.
  - Software can deallocate a line in the L2 by using CLFLUSH, and thus allow for a cache line to be filled with a different location.

Performance monitor event PMCx07F [L2 Fill/Writeback], sub-event bit 1, titled “L2 Writebacks to system“, can be used to indicate whether L2 dirty data was victimized and sent to the disabled memory controller.

The following requirements must be satisfied prior to using the cache as general storage:
- Paging must be disabled.
- MSRC001_0015[INVDWBINVD]=0.
- MSRC001_1020[DIS_SS]=1.
- MSRC001_1021[DIS_SPEC_TLB_RLD]=1. Disable speculative ITLB reloads.
- MSRC001_1022[DIS_SPEC_TLB_RLD]=1. Disable speculative DTLB reloads.
- MSRC001_1022[DIS_CLR_WBTOL2_SMC_HIT]=1.
- MSRC001_1022[DIS_HW_PF]=1.
- MSRC001_1029[ClflushSerialize]=1.
- INVD, and WBINVD must not be used during cache as general storage but may be used when tearing down cache-as-ram for all cores on a node.
- The BIOS must not use 3DNow!™, SSE, or MMX™ instructions, with the exception of the following list: MOVD, MOVQ, MOVDQA, MOVQ2DQ, MOVDQ2Q.
- BIOS must not enable exceptions, page-faults, and other interrupts.
- BIOS must not use software prefetches.

When BIOS is done using the cache as general storage the following steps are followed:
1. An INVD instruction should be executed on each core that used cache as general storage.
2. If DRAM is initialized and there is data in the cache that needs to get moved to main memory, CLFLUSH or WBINVD may be used instead of INVD, but software must ensure that needed data in main memory is not overwritten.
3. Restore the following configuration state:
   - MSRC001_0015[INVDWBINVD].
   - MSRC001_1020[DIS_SS]=0.
   - MSRC001_1021[DIS_SPEC_TLB_RLD]=0.
   - MSRC001_1022[DIS_SPEC_TLB_RLD]=0.
   - MSRC001_1022[DIS_CLR_WBTOL2_SMC_HIT]=0.
   - MSRC001_1022[DIS_HW_PF]=0.
   - MSRC001_1029[ClflushSerialize]=0.

### 2.3.4 BIOS Requirements For 64-Bit Operation

Refer to the AMD64 Architecture Programmer's Manual for a description of the 64-bit mode.

## 2.4 Processor Core

The majority of the behavioral definition of the processor core is specified in the AMD64 Architecture Programmer's Manual. See 1.2 [Reference Documents].

### 2.4.1 Virtual Address Space

The processor supports 48 address bits of virtual memory space (256 terabyte) as indicated by CPUID Fn8000_0008_EAX.

### 2.4.2 Processor Cores and Downcoring

Each processor supports 1 to 4 cores.

- The number of cores supported by the processor is specified by D18F3xE8[CmpCap].
- The core number, CpuCoreNum, is provided to SW running on each core through CPUID Fn0000_0001_EBX[LocalApicId] and APIC20[ApicId]; CpuCoreNum also affects D18F0x68[Cpu1En] and D18F0x168[Cpu3En, Cpu2En].
- The boot core is always the core reporting CpuCoreNum = 0.

### 2.4.3 Access Type Determination

The access type determination and destination affects routing specified in 2.8.4 [Northbridge Routing].

### 2.4.4 System Address Map

The processor defines a reserved memory address region starting at 0000_00FD_0000_0000h and extending up to 0000_0100_0000_0000h. System software must not map memory into this region. Downstream host accesses to the reserved address region results in a page fault. Upstream system device accesses to the reserved address region results in an undefined operation.

#### 2.4.4.1 Memory Access to the Physical Address Space

All memory accesses to the physical address space from a core are sent to its associated northbridge (NB). All memory accesses from a link are routed through the NB.

A core access to physical address space has two important attributes that the CPU must determine before issuing the access to the NB: the cache attribute (e.g., WB, WC, UC; as described in the MTRRs) and the access destination (DRAM or MMIO).

##### 2.4.4.1.1 Determining The Cache Attribute

1. The CPU translates the logical address to a physical address. In that process it determines the initial cache attribute based on the settings of the Page Table Entry PAT bits, [The MTRR Default Memory Type Register (MTRRdefType)] MSR0000_02FF, [The Variable-Size MTRRs (MTRRphysBasen)] MSR0000_020[E,C,A,8,6,4,2,0], and [The Fixed-Size MTRRs] MSR0000_02[6F:68,59,58,50].
2. The ASeg and TSeg SMM mechanisms are then checked in parallel to determine if the initial cache attribute should be overridden (see [The SMM TSeg Base Address Register (SMMAddr)] MSRC001_0112

and [The SMM TSeg Mask Register (SMMMask)] MSRC001_0113). If the address falls within an enabled ASeg/TSeg region, then the final cache attribute is determined as specified in MSRC001_0113.

This mechanism is managed by BIOS and does not require any setup or changes by system software.

#### 2.4.4.1.2 Determining The Access Destination for CPU Accesses

The access destination, DRAM or MMIO, is based on the highest priority of the following ranges that the access falls in:

1. (Lowest priority) Compare against the top-of memory (TOM) registers (see MSRC001_001A, and MSRC001_001D).
2. [The Fixed-Size MTRRs] MSR0000_02[6F:68,59,58,50].
3. The IORRs (see MSRC001_00[18,16] and MSRC001_00[19,17]).
4. TSEG & ASEG (see MSRC001_0112 and MSRC001_0113). (Highest priority)

To determine the access destination, the following steps are taken:

1. The CPU compares the address against [The Top Of Memory Register (TOP_MEM)] MSRC001_001A, and [The Top Of Memory 2 Register (TOM2)] MSRC001_001D, to determine if the default access destination is DRAM or MMIO space.
2. For addresses below 1M byte, the address is then compared against the appropriate Fixed MTRRs to override the default access destination. Each fixed MTRR includes two bits, RdDram and WrDram, that determine the destination based on the access type. See MSR0000_02[6F:68,59,58,50].
3. The CPU then compares the address against the IORRs (MSRC001_00[18,16] and MSRC001_00[19,17]); if it matches, the default access destination is overridden as specified by the IORRs. BIOS can use the IORRs to create an IO hole within a range of addresses that would normally be mapped to DRAM. It can also use the IORRs to re-assert a DRAM destination for a range of addresses that fall within a bigger IO hole that overlays DRAM.
    a) Operating system software never needs to program IORRs to re-map addresses that naturally target DRAM; any such programming is done by BIOS.
4. The ASeg and TSeg SMM mechanisms are then checked in parallel to determine if the destination should be overridden (see MSRC001_0112 and MSRC001_0113). If the address falls within an enabled ASeg/TSeg region, then the destination is determined as specified in MSRC001_0113.

This mechanism is managed by BIOS and does not require any setup or changes by system software.

### 2.4.5 Timers

Each core includes the following timers. These timers do not vary in frequency regardless of the current P-state or C-state.

- [The Time Stamp Counter Register (TSC)] MSR0000_0010; the TSC increments at the rate specified by MSRC001_0015[TscFreqSel].
- The APIC timer (APIC380 and APIC390), which decrements at a rate of 2x CLKIN.

### 2.4.6 Interrupts

#### 2.4.6.1 Local APIC

The local APIC contains logic to receive interrupts from a variety of sources and to send interrupts to other local APICs, as well as registers to control its behavior and report status. Interrupts can be received from:

- IO devices including the FCH (IO APICs)
- Other local APICs (inter-processor interrupts)
- APIC timer
- Thermal events
- Performance counters
- Legacy local interrupts from the FCH (INTR and NMI)
- APIC internal errors

The APIC timer, thermal events, performance counters, local interrupts, and internal errors are all considered local interrupt sources, and their routing is controlled by local vector table entries. These entries assign a message type and vector to each interrupt, allow them to be masked, and track the status of the interrupt.

IO and inter-processor interrupts have their message type and vector assigned at the source and are unaltered by the local APIC. They carry a destination field and a mode bit that together determine which local APIC(s) accepts them. The destination mode (DM) bit specifies if the interrupt request packet should be handled in physical or logical destination mode. If the destination field matches the broadcast value specified by D18F0x68[ApicExtBrdCst], then the interrupt is a broadcast interrupt and is accepted by all local APICs regardless of destination mode.

#### 2.4.6.1.1 ApicId Enumeration Requirements

System hardware and BIOS must ensure that the number of cores per processor (NC) exposed to the operating system by all tables, registers, and instructions across all cores and processors in the system is identical. See2.4.8.1 [Multi-Core Support] to derive NC.

Operating systems are expected to use CPUID Fn8000_0008_ECX[ApicIdCoreIdSize[3:0]], the number of least significant bits in the Initial APIC ID that indicate core ID within a processor, in constructing per-core CPUID masks. (ApicIdCoreIdSize[3:0] determines the maximum number of cores (MNC) that the processor could theoretically support, not the actual number of cores that are actually implemented or enabled on the processor, as indicated by CPUID Fn8000_0008_ECX[NC].) BIOS must use the ApicId MNC rule when assigning [The APIC ID Register] APIC20[ApicId] values as described below.

ApicId MNC rule: The ApicId of core j must be enumerated/assigned as:

ApicId[core=j] = (OFFSET_IDX) * MNC + j

Where "OFFSET_IDX" is an integer offset (0 to N) used to shift up the CPU ApicId values to allow room for IOAPIC devices.

It is recommended that BIOS use the following APIC ID assignments for the broadest operating system support. Given N = (MNC) and M = Number_Of_IOAPICs:

- Assign the core ApicId’s first from 0 to N-1, and the IOAPIC IDs from N to N+(M-1).

#### 2.4.6.1.2 Physical Destination Mode

The interrupt is only accepted by the local APIC whose APIC20[ApicId] matches the destination field of the interrupt. Physical mode allows up to 255 APICs to be addressed individually.

#### 2.4.6.1.3 Logical Destination Mode

A local APIC accepts interrupts selected by [The Logical Destination Register] APICD0 and the destination field of the interrupt using either cluster or flat format as configured by APICE0[Format].

If flat destinations are in use, bits 7-0 of APICD0[Destination] are checked against bits 7-0 of the arriving interrupt's destination field. If any bit position is set in both fields, the local APIC is a valid destination. Flat format allows up to 8 APICs to be addressed individually.

If cluster destinations are in use, bits 7-4 of APICD0[Destination] are checked against bits 7-4 of the arriving interrupt's destination field to identify the cluster. If all of bits 7-4 match, then bits 3-0 of APICD0[Destination] and the interrupt destination are checked for any bit positions that are set in both fields to identify processors within the cluster. If both conditions are met, the local APIC is a valid destination. Cluster format allows 15 clusters of 4 APICs each to be addressed.

#### 2.4.6.1.4 Interrupt Delivery

SMI, NMI, INIT, Startup, and External interrupts are classified as non-vectored interrupts.

When an APIC accepts a non-vectored interrupt, it is handled directly by the processor instead of being queued in the APIC. When an APIC accepts a fixed or lowest-priority interrupt, it sets the bit in [The Interrupt Request Registers] APIC[270:200] corresponding to the vector in the interrupt. For local interrupt sources, this comes from the vector field in that interrupt's local vector table entry. If a subsequent interrupt with the same vector arrives when the corresponding bit in APIC[270:200][RequestBits] is already set, the two interrupts are collapsed into one. Vectors 15-0 are reserved.

#### 2.4.6.1.5 Vectored Interrupt Handling

[The Task Priority Register] APIC80 and [The Processor Priority Register] APICA0 each contain an 8-bit priority divided into a main priority (bits 7-4) and a priority sub-class (bits 3-0). The task priority is assigned by software to set a threshold priority at which the processor is interrupted.

The processor priority is calculated by comparing the main priority (bits 7-4) of APIC80[Priority] to bits 7-4 of the 8-bit encoded value of the highest bit set in [The In-Service Registers] APIC[170:100]. The processor priority is the higher of the two main priorities.

The processor priority is used to determine if any accepted interrupts (indicated by APIC[270:200][RequestBits]) are high enough priority to be serviced by the processor. When the processor is ready to service an interrupt, the highest bit in APIC[270:200][RequestBits] is cleared, and the corresponding bit is set in APIC[170:100][InServiceBits]. The corresponding bit in [The Trigger Mode Registers] APIC[1F0:180] is set if the interrupt is level-triggered and cleared if edge-triggered.

When the processor has completed service for an interrupt, it performs a write to [The End of Interrupt Register] APICB0, clearing the highest bit in APIC[170:100][InServiceBits] and causing the next-highest interrupt to be serviced. If the corresponding bit in APIC[1F0:180][TriggerModeBits] is set, a write to APICB0 is performed on all APICs to complete service of the interrupt at the source.

#### 2.4.6.1.6 Interrupt Masking

Interrupt masking is controlled by the [The Extended APIC Control Register] APIC410. If APIC410[IerCap] is set, [The Interrupt Enable] APIC[4F0:480] are used to mask interrupts. Any bit in APIC[4F0:480][InterruptEnableBits] that is clear indicates the corresponding interrupt is masked. A masked interrupt is not serviced and

the corresponding bit in APIC[270:200][RequestBits] remains set.

#### 2.4.6.1.7 Spurious Interrupts

In the event that the task priority is set to or above the level of the interrupt to be serviced, the local APIC delivers a spurious interrupt vector to the processor, as specified by [The Spurious Interrupt Vector Register] APICF0. APIC[170:100] is not changed and no write to APICB0 occurs.

#### 2.4.6.1.8 Spurious Interrupts Caused by Timer Tick Interrupt

A typical interrupt is asserted until it is serviced. An interrupt is deasserted when software clears the interrupt status bit within the interrupt service routine. Timer tick interrupt is an exception, since it is deasserted regardless of whether it is serviced or not.

The processor is not always able to service interrupts immediately (i.e. when interrupts are masked by clearing EFLAGS.IM).

If the processor is not able to service the timer tick interrupt for an extended period of time, the INTR caused by the first timer tick interrupt asserted during that time is delivered to the local APIC in ExtInt mode and latched, and the subsequent timer tick interrupts are lost. The following cases are possible when the processor is ready to service interrupts:

- An ExtInt interrupt is pending, and INTR is asserted. This results in timer tick interrupt servicing. This occurs 50 percent of the time.
- An ExtInt interrupt is pending, and INTR is deasserted. The processor sends the interrupt acknowledge cycle, but when the PIC receives it, INTR is deasserted, and the PIC sends a spurious interrupt vector. This occurs 50 percent of the time.

There is a 50 percent probability of spurious interrupts to the processor.

#### 2.4.6.1.9 Lowest-Priority Interrupt Arbitration

Fixed, remote read, and non-vectored interrupts are accepted by their destination APICs without arbitration.

Delivery of lowest-priority interrupts requires all APICs to arbitrate to determine which one accepts the interrupt. If APICF0[FocusDisable] is clear, then the focus processor for an interrupt always accepts the interrupt. A processor is the focus of an interrupt if it is already servicing that interrupt (corresponding bit in APIC[170:100][InServiceBits] is set) or if it already has a pending request for that interrupt (corresponding bit in APIC[270:200][RequestBits] is set). If APIC410[IerCap] is set the interrupt must also be enabled in APIC[4F0:480][InterruptEnableBits] for a processor to be the focus processor. If there is no focus processor for an interrupt, or focus processor checking is disabled, then each APIC calculates an arbitration priority value, stored in [The Arbitration Priority Register] APIC90, and the one with the lowest result accepts the interrupt.

The arbitration priority value is calculated by comparing APIC80[Priority] with the 8-bit encoded value of the highest bit set in APIC[270:200][RequestBits] (IRRVec) and the 8-bit encoded value of the highest bit set APIC[170:100][InServiceBits] (ISRVec). If APIC410[IerCap] is set the IRRVec and ISRVec are based off the highest enabled interrupt. The main priority bits 7-4 are compared as follows:

If (APIC80[Priority[7:4]] >= IRRVec[7:4] and APIC80[Priority[7:4]] > ISRVec[7:4])
Then APIC90[Priority] = APIC80[Priority]

Else if (IRRVec[7:4] > ISRVec[7:4]) APIC90[Priority] = {IRRVec[7:4],0h}
Else APIC90[Priority] = {ISRVect[7:4],0h}

#### 2.4.6.1.10 Inter-Processor Interrupts

[The Interrupt Command Register Low] APIC300 and [The Interrupt Command Register High] APIC310 provide a mechanism for generating interrupts in order to redirect an interrupt to another processor, originate an interrupt to another processor, or allow a processor to interrupt itself. A write to register APIC300 causes an interrupt to be generated with the properties specified by the APIC300 and APIC310 fields.

#### 2.4.6.1.11 APIC Timer Operation

The local APIC contains a 32-bit timer, controlled by [The Timer Local Vector Table Entry] APIC320, [The Timer Initial Count Register] APIC380, and [The Timer Divide Configuration] APIC3E0. The processor bus clock is divided by the value in APIC3E0[Div] to obtain a time base for the timer. When APIC380[Count] is written, the value is copied into [The Timer Current Count Register] APIC390. APIC390[Count] is decremented at the rate of the divided clock. When the count reaches 0, a timer interrupt is generated with the vector specified in APIC320[Vector]. If APIC320[Mode] specifies periodic operation, APIC390[Count] is reloaded with the APIC380[Count] value, and it continues to decrement at the rate of the divided clock. If APIC320[Mask] is set, timer interrupts are not generated.

#### 2.4.6.1.12 Generalized Local Vector Table

All LVTs (APIC320 through APIC370 and APIC[530:500]) support a generalized message type. The generalized values for MsgType are:
- 000b=Fixed
- 010b=SMI
- 100b=NMI
- 111b=ExtINT

#### 2.4.6.1.13 State at Reset

At power-up or reset, the APIC is hardware disabled (MSR0000_001B[ApicEn]=0) so only SMI, NMI, INIT, and ExtInt interrupts may be accepted.

The APIC can be software disabled through APICF0[APICSWEn]. The software disable has no effect when the APIC is hardware disabled.

When a processor accepts an INIT interrupt, the APIC is reset as at power-up, with the exception that APIC20[ApicId], APIC410, and APIC[530:500] are unaffected.

### 2.4.6.2 System Management Mode (SMM)

System management mode (SMM) is typically used for system control activities such as power management. These activities are typically transparent to the operating system.

#### 2.4.6.2.1 SMM Overview

SMM is entered by a core on the next instruction boundary after a system management interrupt (SMI) is received and recognized. A CPU may be programmed to broadcast a special cycle to the system, indicating that it is entering SMM mode. The core then saves its state into the SMM memory state save area and jumps to the SMI service routine (or SMI handler). The pointer to the SMI handler is specified by MSRs. The code and data

for the SMI handler are stored in the SMM memory area, which may be isolated from the main memory accesses.

The core returns from SMM by executing the RSM instruction from the SMI handler. The core restores its state from the SMM state save area and resumes execution of the instruction following the point where it entered SMM. The core may be programmed to broadcast a special bus cycle to the system, indicating that it is exiting SMM mode.

#### 2.4.6.2.2 Operating Mode and Default Register Values

The software environment after entering SMM has the following characteristics:

- Addressing and operation is in Real mode. A far branch in the SMI handler can only address the lower 1M of memory, unless the SMI handler first switches to protected mode.
- 4-Gbyte segment limits.
- Default 16-bit operand, address, and stack sizes (instruction prefixes can override these defaults).
- Control transfers that do not override the default operand size truncate the EIP to 16 bits.
- Far jumps or calls cannot transfer control to a segment with a base address requiring more than 20 bits, as in Real mode segment-base addressing, unless a change is made into protected mode.
- A20M# is disabled. A20M# assertion or deassertion have no affect during SMM.
- Interrupt vectors use the Real mode interrupt vector table.
- The IF flag in EFLAGS is cleared (INTR is not recognized).
- The TF flag in EFLAGS is cleared.
- The NMI and INIT interrupts are masked.
- Debug register DR7 is cleared (debug traps are disabled).

The SMM base address is specified by [The SMM Base Address Register (SMM_BASE)] MSRC001_0111[SMM_BASE]. Important offsets to the base address pointer are:
- MSRC001_0111[SMM_BASE] + 8000h: SMI handler entry point.
- MSRC001_0111[SMM_BASE] + FE00h - FFFFh: SMM state save area.

#### 2.4.6.2.3 SMI Sources And Delivery

The processor accepts SMIs as link-defined interrupt messages only. The core/node destination of these SMIs is a function of the destination field of these messages. However, the expectation is that all such SMI messages are specified to be delivered globally (to all cores of all nodes).

There are also several local events that can trigger SMIs. However, these local events do not generate SMIs directly. Each of them triggers a programmable IO cycle that is expected to target the SMI command port in the FCH and trigger a global SMI interrupt message back to the coherent fabric.

Local sources of SMI events that generate the IO cycle specified in [The SMI Trigger IO Cycle Register] MSRC001_0056 are:

- In the core, as specified by:
  - [The Machine Check Exception Redirection Register] MSRC001_0022.
  - [The IO Trap Registers (SMI_ON_IO_TRAP_[3:0])] MSRC001_00[53:50].
- All local APIC LVT registers programmed to generate SMIs.

The status for these is stored in SMMFEC4.

#### 2.4.6.2.4 SMM Initial State

After storing the save state, execution starts at MSRC001_0111[SMM_BASE] + 08000h. The SMM initial state is specified in the following table.

**Table 2: SMM initial state**

| Register | SMM Initial State |
|---|---|
| CS | SMM_BASE[19:4] |
| DS | 0000h |
| ES | 0000h |
| FS | 0000h |
| GS | 0000h |
| SS | 0000h |
| General Purpose Registers | Unmodified |
| EFLAGS | 0000_0002h |
| RIP | 0000_0000_0000_8000h |
| CR0 | Bits 0, 2, 3, and 31 cleared (PE, EM, TS, and PG); remainder is unmodified |
| CR4 | 0000_0000_0000_0000h |
| GDTR | Unmodified |
| LDTR | Unmodified |
| IDTR | Unmodified |
| TR | Unmodified |
| DR6 | Unmodified |
| DR7 | 0000_0000_0000_0400h |
| EFER | All bits are cleared except bit 12 (SVME) which is unmodified. |

#### 2.4.6.2.5 SMM Save State

In the following table, the offset field provides the offset from the SMM base address specified by [The SMM Base Address Register (SMM_BASE)] MSRC001_0111.

**Table 3: SMM save state**

| Offset | Size | Contents | | Access |
|---|---|---|---|---|
| FE00h | Word | ES | Selector | Read-only |
| FE02h | 6 Bytes | | reserved | |
| FE08h | Quadword | | Descriptor in memory format | |
| FE10h | Word | CS | Selector | Read-only |
| FE12h | 6 Bytes | | reserved | |
| FE18h | Quadword | | Descriptor in memory format | |
| FE20h | Word | SS | Selector | Read-only |
| FE22h | 6 Bytes | | reserved | |
| FE28h | Quadword | | Descriptor in memory format | |

1. This notation specifies that bit[47] is replicated in each of the 16 MSBs of the DW (sometimes called *sign extended*). The 16 LSBs contain bits[47:32].
2. Only valid for an SMI in guest mode with nested paging enabled.

**Table 3: SMM save state**

| Offset | Size | Contents | | Access |
|---|---|---|---|---|
| FE30h | Word | DS | Selector | Read-only |
| FE32h | 6 Bytes | | reserved | |
| FE38h | Quadword | | Descriptor in memory format | |
| FE40h | Word | FS | Selector | Read-only |
| FE42h | 2 Bytes | | reserved | |
| FE44h | Doubleword | | FS Base {16'b[47], 47:32}[1] | |
| FE48h | Quadword | | Descriptor in memory format | |
| FE50h | Word | GS | Selector | Read-only |
| FE52h | 2 Bytes | | reserved | |
| FE54h | Doubleword | | GS Base {16'b[47], 47:32}[1] | |
| FE58h | Quadword | | Descriptor in memory format | |
| FE60h | 4 Bytes | GDTR | reserved | Read-only |
| FE64h | Word | | Limit | |
| FE66h | 2 Bytes | | reserved | |
| FE68h | Quadword | | Descriptor in memory format | |
| FE70h | Word | LDTR | Selector | Read-only |
| FE72h | Word | | Attributes | |
| FE74h | Doubleword | | Limit | |
| FE78h | Quadword | | Base | |
| FE80h | 4 Bytes | IDTR | reserved | Read-only |
| FE84h | Word | | Limit | |
| FEB6h | 2 Bytes | | reserved | |
| FE88h | Quadword | | Base | |
| FE90h | Word | TR | Selector | Read-only |
| FE92h | Word | | Attributes | |
| FE94h | Doubleword | | Limit | |
| FE98h | Quadword | | Base | |
| FEA0h | Quadword | IO_RESTART_RIP | | Read-only |
| FEA8h | Quadword | IO_RESTART_RCX | | |
| FEB0h | Quadword | IO_RESTART_RSI | | |
| FEB8h | Quadword | IO_RESTART_RDI | | |
| FEC0h | Doubleword | [The SMM IO Trap Offset] SMMFEC0 | | Read-only |
| FEC4 | Doubleword | [The Local SMI Status] SMMFEC4 | | Read-only |
| FEC8h | Byte | [The SMM IO Restart Byte] SMMFEC8 | | Read-write |
| FEC9h | Byte | [The Auto Halt Restart Offset] SMMFEC9 | | Read-write |
| FECAh | Byte | [The NMI Mask] SMMFECA | | Read-write |

1. This notation specifies that bit[47] is replicated in each of the 16 MSBs of the DW (sometimes called *sign extended*). The 16 LSBs contain bits[47:32].
2. Only valid for an SMI in guest mode with nested paging enabled.

**Table 3: SMM save state**

| Offset | Size | Contents | Access |
|---|---|---|---|
| FECBh | 5 Bytes | reserved | |
| FED0h | Quadword | EFER | Read-only |
| FED8h | Quadword | SVM State | Read-only |
| FEE0h | Quadword | Guest VMCB physical address | Read-only |
| FEE8h | Quadword | SVM Virtual Interrupt Control | Read-only |
| FEF0h | 16 Bytes | reserved | |
| FEFCh | Doubleword | [The SMM-Revision Identifier] SMMFEFC | Read-only |
| FF00h | Doubleword | [The SMM Base Address Register (SMM_BASE)] SMMFF00 | Read-write |
| FF04h | 28 Bytes | reserved | |
| FF20h | Quadword | Guest PAT | Read-only |
| FF28h | Quadword | Host EFER[2] | |
| FF30h | Quadword | Host CR4[2] | |
| FF38h | Quadword | Nested CR3[2] | |
| FF40h | Quadword | Host Cr0[2] | |
| FF48h | Quadword | CR4 | |
| FF50h | Quadword | CR3 | |
| FF58h | Quadword | CR0 | |
| FF60h | Quadword | DR7 | Read-only |
| FF68h | Quadword | DR6 | |
| FF70h | Quadword | RFLAGS | Read-write |
| FF78h | Quadword | RIP | Read-write |
| FF80h | Quadword | R15 | |
| FF88h | Quadword | R14 | |
| FF90h | Quadword | R13 | |
| FF98h | Quadword | R12 | |
| FFA0h | Quadword | R11 | |
| FFA8h | Quadword | R10 | |
| FFB0h | Quadword | R9 | |
| FFB8h | Quadword | R8 | |

1. This notation specifies that bit[47] is replicated in each of the 16 MSBs of the DW (sometimes called *sign extended*). The 16 LSBs contain bits[47:32].
2. Only valid for an SMI in guest mode with nested paging enabled.

**Table 3: SMM save state**

| Offset | Size | Contents | Access |
|---|---|---|---|
| FFC0h | Quadword | RDI | Read-write |
| FFC8h | Quadword | RSI | |
| FFD0h | Quadword | RBP | |
| FFD8h | Quadword | RSP | |
| FFE0h | Quadword | RBX | |
| FFE8h | Quadword | RDX | |
| FFF0h | Quadword | RCX | |
| FFF8h | Quadword | RAX | |

1. This notation specifies that bit[47] is replicated in each of the 16 MSBs of the DW (sometimes called *sign extended*). The 16 LSBs contain bits[47:32].
2. Only valid for an SMI in guest mode with nested paging enabled.

The SMI save state includes most of the integer execution unit. Not included in the save state are: the floating point state, MSRs, and CR2. In order to be used by the SMI handler, these must be saved and restored. The save state is the same, regardless of the operating mode (32-bit or 64-bit).

The following are some offsets in the SMM save state area. The mnemonic for each offset is in the form SMMxxxx, where xxxx is the offset in the save state.

### SMMFEC0 SMM IO Trap Offset

If the assertion of SMI is recognized on the boundary of an IO instruction, [The SMM IO Trap Offset] SMMFEC0 contains information about that IO instruction. For example, if an IO access targets an unavailable device, the system can assert SMI and trap the IO instruction. SMMFEC0 then provides the SMI handler with information about the IO instruction that caused the trap. After the SMI handler takes the appropriate action, it can reconstruct and then re-execute the IO instruction from SMM. Or, more likely, it can use [The SMM IO Restart Byte] SMMFEC8, to cause the core to re-execute the IO instruction immediately after resuming from SMM.

| Bits | Description |
|---|---|
| 31:16 | **Port: trapped IO port address**. Read-only. This provides the address of the IO instruction. |
| 15:12 | **BPR: IO breakpoint match**. Read-only. |
| 11 | **TF: EFLAGS TF value**. Read-only. |
| 10:7 | Reserved. |
| 6 | **SZ32: size 32 bits**. Read-only. 1=Port access was 32 bits. |
| 5 | **SZ16: size 16 bits**. Read-only.1= Port access was 16 bits. |
| 4 | **SZ8: size 8 bits**. Read-only. 1=Port access was 8 bits. |
| 3 | **REP: repeated port access**. Read-only. |
| 2 | **STR: string-based port access**. Read-only. |
| 1 | **V: IO trap word valid**. Read-only. 1=The core entered SMM on an IO instruction boundary; all information in this offset is valid. 0=The other fields of this offset are not valid. |
| 0 | **RW: port access type**. Read-only. 0=IO write (OUT instruction). 1=IO read (IN instruction). |

### SMMFEC4 Local SMI Status

This offset stores status bits associated with SMI sources local to the core. For each of these bits, 1=The associated mechanism generated an SMI.

| Bits | Description |
|---|---|
| 31:18 | Reserved. |
| 17 | **SmiSrcLvtExt: SMI source LVT extended entry**. Read-only. This bit is associated with the SMI sources specified in [The Extended Interrupt [3:0] Local Vector Table Registers] APIC[530:500]. |
| 16 | **SmiSrcLvtLcy: SMI source LVT legacy entry**. Read-only. This bit is associated with the SMI sources specified by the non-extended LVT entries of the APIC. |
| 15:9 | Reserved. |
| 8 | **MceRedirSts: machine check exception redirection status**. Read-only. This bit is associated with the SMI source specified in [The Machine Check Exception Redirection Register] MSRC001_0022[RedirSmiEn]. |
| 7:4 | Reserved. |
| 3:0 | **IoTrapSts: IO trap status**. Read-only. Each of these bits is associated with each of the respective SMI sources specified in [The IO Trap Registers (SMI_ON_IO_TRAP_[3:0])] MSRC001_00[53:50]. |

### SMMFEC8 SMM IO Restart Byte

00h on entry into SMM.

If the core entered SMM on an IO instruction boundary, the SMI handler may write this to FFh. This causes the core to re-execute the trapped IO instruction immediately after resuming from SMM. The SMI handler should only write to this byte if SMMFEC0[V]=1; otherwise, the behavior is undefined.

If a second SMI is asserted while a valid IO instruction is trapped by the first SMI handler, the CPU services the second SMI prior to re-executing the trapped IO instruction. SMMFEC0[V]=0 during the second entry into SMM, and the second SMI handler must not rewrite this byte.

If there is a simultaneous SMI IO instruction trap and debug breakpoint trap, the processor first responds to the SMI and postpones recognizing the debug exception until after resuming from SMM. If debug registers other than DR6 and DR7 are used while in SMM, they must be saved and restored by the SMI handler. If [The SMM IO Restart Byte] SMMFEC8, is set to FFh when the RSM instruction is executed, the debug trap does not occur until after the IO instruction is re-executed.

| Bits | Description |
|---|---|
| 7:0 | **RST: SMM IO Restart Byte**. Read-write. |

**SMMFEC9 Auto Halt Restart Offset**

| Bits | Description |
|---|---|
| 7:1 | Reserved. |
| 0 | **HLT: halt restart**. Read-write.<br>IF (SmmEntry) THEN<br>Upon SMM entry, this bit indicates whether SMM was entered from the halt state. 0=Entered SMM on a normal x86 instruction boundary. 1=Entered SMM from the halt state.<br>ELSE<br>Before returning from SMM, this bit can be written by the SMI handler to specify whether the return from SMM should take the processor back to the halt state or to the instruction-execution state specified by the SMM state save area (normally, the instruction after the halt). 0=Return to the instruction specified in the SMM save state. 1=Return to the halt state. If the return from SMM takes the processor back to the halt state, the HLT instruction is not refetched and re-executed. However, the halt special bus cycle is broadcast and the processor enters the halt state.<br>ENDIF. |

**SMMFECA NMI Mask**

| Bits | Description |
|---|---|
| 7:1 | Reserved. |
| 0 | **NmiMask**. Read-write. Specifies whether NMI was masked upon entry to SMM. 0=NMI not masked. 1=NMI masked. |

**SMMFED8 SMM SVM State**

This offset stores the SVM state of the processor upon entry into SMM.

| Bits | Description |
|---|---|
| 63:4 | Reserved. |
| 3 | **HostEflagsIf: host Eflags IF**. Read-only. |
| 2:0 | **SvmState**. Read-only.<br>Bits — Definition<br>000b — SMM entered from a non-guest state.<br>001b — Reserved.<br>010b — SMM entered from a guest state.<br>101b-011b — Reserved.<br>110b — SMM entered from a guest state with nested paging enabled.<br>111b — Reserved. |

**SMMFEFC SMM-Revision Identifier**

SMM entry state: 0003_0064h

| Bits | Description |
|---|---|
| 31:18 | Reserved. |
| 17 | **BRL**. Read-only. Base relocation supported. |
| 16 | **IOTrap**. Read-only. IO trap supported. |
| 15:0 | **Revision**. Read-only. |

**SMMFF00 SMM Base Address Register (SMM_BASE)**

| Bits | Description |
|---|---|
| 31:0 | See: MSRC001_0111[31:0]. This offset is loaded with the contents of MSRC001_0111. |

#### 2.4.6.2.6 Exceptions and Interrupts in SMM

When SMM is entered, the CPU masks INTR, NMI, SMI, INIT, and A20M interrupts. The CPU clears the IF flag to disable INTR interrupts. To enable INTR interrupts within SMM, the SMM handler must set the IF flag to 1. A20M is disabled so that address bit 20 is never masked when in SMM.

Generating an INTR interrupt can be used for unmasking NMI interrupts in SMM. The CPU recognizes the assertion of NMI within SMM immediately after the completion of an IRET instruction. Once NMI is recognized within SMM, NMI recognition remains enabled until SMM is exited, at which point NMI masking is restored to the state it was in before entering SMM.

While in SMM, the CPU responds to the DBREQ and STPCLK interrupts, as well as to all exceptions that may be caused by the SMI handler.

#### 2.4.6.2.7 The Protected ASeg and TSeg Areas

These ranges are controlled by MSRC001_0112 and MSRC001_0113.

#### 2.4.6.2.8 SMM Special Cycles

Special cycles can be initiated on entry and exit from SMM to acknowledge to the system that these transitions are occurring. These are controlled by MSRC001_0015[SMISPCYCDIS, RSMSPCYCDIS].

#### 2.4.6.2.9 Locking SMM

The SMM registers (MSRC001_0112 and MSRC001_0113) can be locked from being altered by setting MSRC001_0015[SmmLock]. BIOS can lock the SMM registers after initialization to prevent unexpected changes to these registers.

### 2.4.7 Secure Virtual Machine Mode (SVM)

Support for SVM mode is indicated by CPUID Fn8000_0001_ECX[SVM]. If SVM is supported, then the DEV registers starting at D18F3xF0 are visible.

#### 2.4.7.1 BIOS support for SVM Disable

BIOS should include the following user setup options to enable and disable AMD Virtualization™ technology.

- Enable AMD Virtualization™.
  - MSRC001_0114[Svm_Disable] = 0.
  - MSRC001_0114[Lock] = 1.
  - MSRC001_0118[SvmLockKey] = 0000_0000_0000_0000h.
- Disable AMD Virtualization™.
  - MSRC001_0114[Svm_Disable]=1.
  - MSRC001_0114[Lock]=1.
  - MSRC001_0118[SvmLockKey] = 0000_0000_0000_0000h.

BIOS may also include the following user setup option to disable AMD Virtualization™ technology.

- Disable AMD Virtualization™, with a user supplied key.
  - MSRC001_0114[Svm_Disable]=1.
  - MSRC001_0114[Lock]=1.
  - MSRC001_0118[SvmLockKey] programmed with value supplied by user. This value should be NVRAM.

### 2.4.8 CPUID Instruction

The CPUID instruction provides data about the features supported by the processor. See 3.19 [CPUID Instruction Registers].

#### 2.4.8.1 Multi-Core Support

There are two methods for determining multi-core support. A recommended mechanism is provided and a legacy method is also available for existing operating systems. System software should use the correct architectural mechanism to detect the number of physical cores by observing CPUID Fn8000_0008_ECX[NC]. The legacy method utilizes the CPUID Fn0000_0001_EBX[LogicalProcessorCount].

## 2.5 Power Management

The processor supports many power management features in a variety of systems. Table 4 provides a summary of ACPI states and power management features and indicates whether they are supported.

**Table 4: Power management support**

| ACPI/Power Management State | Supported | Description |
|---|---|---|
| G0/S0/C0: Working | Yes | |
| G0/S0/C0: Core P-state transitions | Yes | 2.5.3.1 [Core P-states] |
| G0/S0/C0: NB P-state transitions | Yes | 2.5.4.2 [NB Clock Ramping] |
| G0/S0/C0: Hardware thermal control (HTC) | Yes | 2.10.3.1 [PROCHOT_L and Hardware Thermal Control (HTC)] |
| G0/S0/C0: Software thermal control (STC) | No | |
| G0/S0/C0: Thermal clock throttling (SMC controlled) | No | |
| G0/S0: Low power C-states | Yes | 2.5.3.2 [C-states] and 2.5.1.4.2 [Alternate Low Power Voltages] |
| G1/S1: Stand By (Powered On Suspend) | No | |

**Table 4: Power management support**

| ACPI/Power Management State | Supported | Description |
|---|---|---|
| G1/S3: Stand By (Suspend to RAM) | Yes | 2.5.7.1.1 [ACPI Suspend to RAM State (S3)] |
| G1/S4, S5: Hibernate (Suspend to Disk), Shut Down (Soft Off) | Yes | |
| G3 Mechanical Off | Yes | |
| Parallel VID Interface | No | 2.5.1 [Processor Power Planes And Voltage Control] |
| Serial VID Interface | Yes | |
| Dual-plane systems | Yes | |

### 2.5.1 Processor Power Planes And Voltage Control

Refer to the AMD Family 12h Processor Electrical Data Sheet, #41609 for power plane electrical requirements and definitions.

#### 2.5.1.1 Internal VID Registers

The registers within the processor that contain VID fields all use 7-bit VID encodings. See AMD Voltage Regulator Specification, #40182.

##### 2.5.1.1.1 VID Encodings

The VID encoding to voltage translations for all VID codes are defined by the AMD Voltage Regulator Specification, #40182.

The boot voltage is 1.0 volts.

##### 2.5.1.1.2 MinVid and MaxVid Check

Hardware limits the minimum and maximum VID code that is sent to the voltage regulator. The allowed limits of MinVid and MaxVid are provided in MSRC001_0071. Prior to generating VID-change commands to SVI, the processor filters the InputVid value to the OutputVid as follows (higher VID codes correspond to lower voltages and lower VID codes correspond to higher voltages):

- If InputVid < MaxVid, OutputVid=MaxVid.
  - Else if (InputVid > MinVid) & (MinVid != 00h), OutputVid=MinVid.
  - Else OutputVid=InputVid.

This filtering is applied regardless of the source of the VID-change command.

#### 2.5.1.2 Serial VID Interface

The processor includes an interface, intended to control external voltage regulators, called the serial VID interface (SVI). Refer to the AMD Voltage Regulator Specification, #40182 for details. The frequency of the SVC is controlled by D18F3xA0[SviHighFreqSel].

#### 2.5.1.3 BIOS Requirements for Power Plane Initialization

1. Initialize D18F3xD8[VSRampSlamTime].
2. Configure D18F3xA0[PsiVidEn, PsiVid] and D18F3x128[NbPsiVidEn, NbPsiVid]. See 2.5.1.4.1 [PSI_L Bit].
3. Program D18F3xDC[NbPs0Vid] = FCRxFE00_6000[NbPs0Vid] - 1.

4. Program D18F3xDC[NbPs0Vid] = FCRxFE00_6000[NbPs0Vid].
5. 2.11.4.2 [Link Configuration and Initialization] may be executed to enable links that operate at 2.5 Gb/s.
6. Request the lowest valid voltage specified by D18F3x15C using D0F0x64_x6A and D0F0x64_x6B. See 2.5.1.5.2 [Software-Initiated Voltage Transitions].
7. After 2.9.3.7 [DRAM Training] is complete, the following actions may be taken in any order:
   - If (GpuEnabled), BIOS requests the highest valid voltage specified by D18F3x15C. BIOS makes this requests using GMMx770 and GMMx774 before video BIOS initialization. See 2.5.1.5.2 [Software-Initiated Voltage Transitions].
   - If the links need to be configured to operate at 5.0Gb/s,
     a. Request the voltage specified by FCRxFE00_4036[PcieGen2Vid] using D0F0x64_x6A and D0F0x64_x6B. See 2.5.1.5.2 [Software-Initiated Voltage Transitions].
     b. Execute 2.11.4.2 [Link Configuration and Initialization] enable links to operate at 5.0 Gb/s.
8. If no links are operating at 5.0 Gb/s, request the lowest valid voltage specified by D18F3x15C using D0F0x64_x6A and D0F0x64_x6B.

### 2.5.1.4 Low Power Features

#### 2.5.1.4.1 PSI_L Bit

The processor supports additional system power savings through the use of a low-voltage state indicator, the PSI_L bit. The PSI_L bit in the data field of the SVI command can be used by the voltage regulator to place itself into a more power efficient mode. The PSI_L bit can be controlled independently for the VDD and VDDNB planes. The PSI_L bit is enabled through D18F3xA0[PsiVidEn] and D18F3x128[NbPsiVidEn]. Once enabled, the state of the PSI_L bit is controlled by D18F3xA0[PsiVid] and D18F3x128[NbPsiVid].

##### 2.5.1.4.1.1 BIOS Requirements for PSI_L

Enabling PSI_L for the VDD and VDDNB planes depends on the voltage regulator and is therefore system specific. Depending on the voltage regulator being used, AMD recommends one of the following methods:

- PSI_L always clear:
  - To clear PSI_L for the VDD plane, program D18F3xA0[PsiVidEn]=1 and D18F3xA0[PsiVid]=0. To clear PSI_L for the VDDNB plane, program D18F3x128[NbPsiVidEn]=1 and D18F3x128[NbPsiVid]=0.

- PSI_L always set:
  - To set PSI_L for the VDD plane, program D18F3xA0[PsiVidEn]=0. To set PSI_L for the VDDNB plane, program D18F3x128[NbPsiVidEn]=0.

- PSI_L set/clear based on current requirements (VDD only):
  - If the voltage regulator requires that PSI_L be set or cleared at a certain current level, BIOS uses the following pseudo-code:

```
psi_vrm_current = current at which the regulator allows PSI_L;
psi_inrush_current = inrush current during a voltage transition;
psi_state_found = false;

for (each valid P-state starting with P0) {
    pstate_current = MSRC001_00[6B:64][IddDiv, IddVal];
    pstate_voltage = MSRC001_00[6B:64][CpuVid];
```

```
        if (current P-state is P0) {
            prev_voltage = 7Fh;
        } else {
            prev_voltage = MSRC001_00[6B:64][CpuVid] of previous P-state;
        }

        if (current P-state is last valid P-state) {
            if (D18F4x1AC[PkgC6Cap] == 1) {
                next_pstate_current = 0;
            } else {
                next_pstate_current = 4 amps;
            }
        } else {
            next_pstate_current = psi_inrush_current +
            MSRC001_00[6B:64][IddDiv, IddVal] of the next P-state;
        }

        if ((pstate_current <= psi_vrm_current) &&
            (next_pstate_current <= psi_vrm_current) &&
            (pstate_voltage != prev_voltage)){
            program D18F3xA0[PsiVid] = pstate_voltage;
            program D18F3xA0[PsiVidEn] = 1;
            psi_state_found = true;
            break;
        }
    }

    if ((psi_state_found == false) &&
        (D18F4x1AC[PkgC6Cap] == 1) &&
        (4 amps <= psi_vrm_current)) {
        program D18F3xA0[PsiVid] = D18F3x128[C6Vid];
        program D18F3xA0[PsiVidEn] = 1;
    }
```

Please contact your voltage regulator vendor and your AMD representative to determine the best method. This section uses hardware P-state numbering. See 2.5.3.1.2.2 [Hardware P-state Numbering].

#### 2.5.1.4.2 Alternate Low Power Voltages

In order to save power, voltages lower than those normally used may be applied to the VDD or VDDNB power planes while the processor is in a C-state or the GPU is idle.

- C6Vid (D18F3x128[C6Vid]): C6Vid specifies a VDD voltage that does not retain the CPU caches or the cores' microarchitectural state, nor allows for execution. As a result, hardware flushes caches and saves the cores' microarchitectural state to DRAM before transitioning to C6Vid. See 2.5.3.2.3.4 [Package C6 (PC6) State].
- GnbIdleAdjustVid (FCRxFE00_705F[GnbIdleAdjustVid]): GnbIdleAdjustVid specifies a voltage offset that is subtracted from the VDDNB voltage. The resulting voltage allows all components on VDDNB to function if the GPU is clock or power gated and incapable of causing voltage transients due to changes in activity. GnbIdleAdjustVid is enabled using FCRxFF30_0191[GfxIdleVoltChgMode, GfxIdleVoltChgEn]. See 2.5.5.4 [GPU and Root Complex Power Gating]. GnbIdleAdjustVid is never applied when the internal GPU

is disabled (see GpuEnabled).

#### 2.5.1.4.3 Power Gating

The processor can remove power from an individual core, the GMC, or the GFX link core. This is referred to as power gating. Gating power to a subcomponent causes its internal microarchitectural state and, if applicable, any data in its caches to be lost. When entering a power gated state, hardware saves any needed data, either internally or to DRAM, and flushes caches. When exiting a power gated state, hardware performs any required resets and restores any needed data. See 2.5.3.2.3.2 [Core C6 (CC6) State] and 2.5.5.4 [GPU and Root Complex Power Gating].

### 2.5.1.5 Voltage Transitions

The processor supports dynamic voltage transitions on the VDD and VDDNB planes. These transitions are controlled by hardware. VDD and VDDNB voltage levels may be transitioned during state changes involving reset, boot, P-state changes, C-state changes, and stop-grant. In all cases, the voltage is *slammed*; this means that the VID code passed to the voltage regulator changes from the old value to the new value without stepping through intermediate values. The voltage regulator ramps the voltage directly from the starting voltage to the final voltage. See the AMD Voltage Regulator Specification, #40182.

If a voltage increase is requested, the processor waits the amount of time specified by D18F3xD8[VSRampSlamTime] before sending any additional voltage change requests to the voltage regulator or before beginning a frequency transition. If a voltage decrease is requested, the processor does not wait any time before sending additional voltage changes or beginning frequency changes. The processor continues execution of code during voltage changes when in the C0 state.

#### 2.5.1.5.1 Hardware-Initiated Voltage Transitions

When software requests any of the following power state changes, or hardware determines that any of the following state changes are necessary, hardware coordinates the necessary voltage changes:

- VDD:
  - Core P-state transition. See 2.5.3.1 [Core P-states].
  - Package C-state transition. See 2.5.3.2 [C-states].
  - S-state transition. See 2.5.7.1 [S-states].

- VDDNB:
  - NB P-state transition. See 2.5.4.1 [NB P-states].
  - DPM state transition. See 2.5.5.1 [Dynamic Power Management (DPM)].
  - S-state transition. See 2.5.7.1 [S-states].

#### 2.5.1.5.2 Software-Initiated Voltage Transitions

Software can request voltage changes on the VDDNB power plane using the following control/status register pairs:

- D0F0x64_x6A and D0F0x64_x6B

GMMx770 and GMMx774

The voltage requests from each register pair are considered independently by hardware when taking voltage plane dependencies into account (see 2.5.2.2 [Dependencies Between Subcomponents on VDDNB]). To make a voltage change request, software uses the following sequence:

1. Ensure VoltageChangeEn==1 in the control register. If software needs to program VoltageChangeEn=1, software must perform this register write independently of the writes in the following steps.
2. Program VoltageLevel to the desired voltage and toggle VoltageChangeReq in the control register.
3. The voltage change is complete when VoltageChangeReq in the control register is equal to VoltageChange-Ack in the status register.

Software can force a VDDNB voltage change using D0F0x64_x6A and GMMx770. To do so, software programs VoltageForceEn=1 in the respective register before toggling VoltageChangeReq when making a voltage change request. If this is done, the voltage requested overrides any other VDDNB voltage requests made by software or DPM state transitions when determining voltage plane dependencies (see 2.5.2.2 [Dependencies Between Subcomponents on VDDNB]). NB P-state transitions still request voltage transitions as normal when software forces a voltage change using this mechanism. If software forces a voltage change using both D0F0x64_x6A and GMMx770, the voltage requested in D0F0x64_x6A takes precedence.

The following registers also cause VDDNB voltage transitions:

- D18F3xDC[NbPs0Vid] when in NBP0: See 2.5.4.1 [NB P-states].

### 2.5.2 Frequency and Voltage Domain Dependencies

#### 2.5.2.1 Dependencies Between Cores

Whenever a P-state or C-state is requested on a core (see 2.5.3.1 [Core P-states] and 2.5.3.2 [C-states]), hardware must take frequency and voltage domain dependencies into account when determining whether to make the requested state change. Whenever software requests different power management states for the cores on a multi-core processor, the processor decides which state to target based on a set of policy controls, described below.

- If multiple cores request different P-states, frequency changes independently on each core based on the P-state requested by that core. The VDD voltage is determined by the highest-performance P-state requested on any core. See 2.5.3.1 [Core P-states].
- If some cores request non-C0 C-states while other cores are in C0, core C-state actions are taken independently by the cores requesting those states. Package C-state actions are not taken. The VDD voltage is determined by the highest-performance P-state requested on any core.
- If all cores request non-C0 C-states, core C-state actions are taken independently by each core and package C-state actions are determined by the shallowest C-state request made by any core.

#### 2.5.2.2 Dependencies Between Subcomponents on VDDNB

Many subcomponents of the processor including the NB, the GPU, and the root complex reside on the VDDNB power plane. Hardware must take voltage domain dependencies into account when determining whether to make a voltage change requested by one of the subcomponents. Whenever a state transition occurs that causes a voltage change request (see 2.5.1.5.1 [Hardware-Initiated Voltage Transitions]), or software makes a voltage change request (see 2.5.1.5.2 [Software-Initiated Voltage Transitions]), the VDDNB voltage requested by the processor is determined by the highest voltage (lowest VID) request made by any of the subcomponents or by software. In addition, software can force VDDNB voltage changes. See 2.5.1.5.2 [Software-Initiated Voltage Transitions].

### 2.5.3 CPU Power Management

#### 2.5.3.1 Core P-states

Core P-states are operational performance states characterized by a unique combination of core frequency and voltage. The processor supports up to 8 core P-states, specified in MSRC001_00[6B:64]. Out of reset, the voltage and frequency of the cores is specified by MSRC001_0071[StartupPstate].

Support for dynamic core P-state changes is indicated by more than one enabled selection in MSRC001_00[6B:64][PstateEn]. All FID and DID parameters for equivalent P-states must be programmed to equivalent values for all cores. For examples, P0 on core0 must have the same FID and DID values as P0 on core1, P1 on core0 must have the same FID and DID values as P1 on core1, and so on. Refer to MSRC001_00[6B:64] for further details on programming requirements. The COF for core P-states is a function of the FID and the DID. See MSRC001_00[6B:64][CpuFid, CpuDid] for more details.

Software requests core P-state changes for each core independently using the hardware P-state control mechanism (a.k.a. fire and forget). Support for hardware P-state control is indicated by CPUID Fn8000_0007_EDX[HwPstate]=1b. P-state transitions using the hardware P-state control mechanism are not allowed until the P-state initialization requirements defined in 2.5.3.1.7 [BIOS Requirements for Core P-State Initialization and Transitions] are complete.

##### 2.5.3.1.1 Core Performance Boost (CPB)

Core performance boost (CPB) allows the processor to provide maximum performance while remaining within a specified power delivery and removal envelope. CPB hardware dynamically monitors processor activity and generates an approximation of power consumption. If power consumption exceeds a defined power limit, a P-state limit is applied by CPB hardware to reduce power consumption. CPB ensures that average power consumption over a thermally significant time period remains at or below the defined power limit. This allows P-states to be defined with higher frequencies and voltages than could be used without CPB. These P-states are referred to as boosted P-states.

- Support for CPB is specified by CPUID Fn8000_0007_EDX[CPB].
- CPB is enabled if MSRC001_0015[CpbDis]==0 and D18F4x15C[BoostSrc]!=0. CPB is disabled if either MSRC001_0015[CpbDis]==1 or D18F4x15C[BoostSrc]==0.
  - If CPB is disabled using MSRC001_0015[CpbDis], a P-state limit is applied to the core on which CPB was disabled. This P-state limit restricts that core to the highest performance non-boosted P-state.
  - If CPB is disabled using D18F4x15C[BoostSrc], a P-state limit is applied to all cores in the system. This P-state limit restricts the core(s) to the highest performance non-boosted P-state.
- All P-states, both boosted and non-boosted, are specified in MSRC001_00[6B:64].
- The number of boosted P-states is specified by D18F4x15C[NumBoostStates].
  - The number of boosted P-states may vary from product to product.
- All boosted P-states are always higher performance than non-boosted P-states.
- To ensure proper OS operation, boosted P-states should be hidden from the operating system. BIOS should not provide ACPI _PSS entries for boosted P-states. See 2.5.3.1.9.2 [_PSS (Performance Supported States)].
- The lowest-performance P-state CPB limits the processor to is the highest-performance non-boosted P-state.

##### 2.5.3.1.2 Core P-state Naming and Numbering

Since the number of boosted P-states may vary from product to product, the mapping between MSRC001_00[6B:64] and the indices used to request P-state changes or status also varies. In order to clarify this, two different numbering schemes are used.

#### 2.5.3.1.2.1 Software P-state Numbering

When referring to software P-state numbering, the following naming convention is used:

- Non-boosted P-states are referred to as P0, P1, etc.
  - P0 is the highest power, highest performance, non-boosted P-state.
  - Each ascending P-state number represents a lower-power, lower performance non-boosted P-state than the prior P-state number.
- Boosted P-states are referred to as Pb0, Pb1, etc.
  - Pb0 is the highest-performance, highest-power boosted P-state.
  - Each higher numbered boosted P-state represents a lower-power, lower-performance boosted P-state.

For example, if D18F4x15C[NumBoostStates] contains the values shown below, then the P-states would be named as follows:

**Table 5: Software P-state numbering example**

| D18F4x15C[NumBoostStates]=1 | | D18F4x15C[NumBoostStates]=3 | |
|---|---|---|---|
| **P-state Name** | **MSR Address** | **P-state Name** | **MSR Address** |
| Pb0 | MSRC001_0064 | Pb0 | MSRC001_0064 |
| P0 | MSRC001_0065 | Pb1 | MSRC001_0065 |
| P1 | MSRC001_0066 | Pb2 | MSRC001_0066 |
| P2 | MSRC001_0067 | P0 | MSRC001_0067 |
| P3 | MSRC001_0068 | P1 | MSRC001_0068 |
| P4 | MSRC001_0069 | P2 | MSRC001_0069 |
| P5 | MSRC001_006A | P3 | MSRC001_006A |
| P6 | MSRC001_006B | P4 | MSRC001_006B |

All sections and register definitions use software P-state numbering unless otherwise specified.

#### 2.5.3.1.2.2 Hardware P-state Numbering

When referring to hardware P-state numbering, the following naming convention is used:

- All P-states are referred to as P0, P1, etc.
  - P0 is the highest power, highest-performance P-state, regardless of whether it is a boosted P-state or a non-boosted P-state.
  - Each ascending P-state number represents a lower-power, lower-performance P-state, regardless of whether it is a boosted P-state or not.

### 2.5.3.1.3 Core P-state Control

Core P-states are dynamically controlled by software and are exposed through ACPI objects (see 2.5.3.1.9 [ACPI Processor P-State Objects]). Software requests a core P-state change by writing a 3 bit index corresponding to the desired non-boosted P-state number to [The P-State Control Register] MSRC001_0062 of the appropriate core. For example, to request P3 for core 0 software would write 011b to core 0's MSRC001_0062[PstateCmd]. Boosted P-states may not be directly requested by software. Whenever software requests the P0 state on a processor that supports CPB (i.e. writes 000b to MSRC001_0062[PstateCmd]), hardware dynamically places that core into the highest-performance P-state possible as determined by CPB. See 2.5.3.1.1 [Core Performance Boost (CPB)].

Hardware sequences the frequency and voltage changes necessary to complete a P-state transition as specified by 2.5.3.1.6 [Core P-state Transition Behavior] with no additional software interaction required. Hardware also coordinates frequency and voltage changes when differing P-state requests are made on cores that share a frequency or voltage plane. See 2.5.2 [Frequency and Voltage Domain Dependencies] for details about hardware coordination.

Core P-states are changed without interacting with an external chipset. However, the chipset is notified of core P-state changes by the P-state special cycle if MSRC001_001F[EnaPStateSpCyc]=1. This message is sent regardless of whether the change is to or from a boosted P-state or a non-boosted P-state.

**Table 6: P-state control example**

| D18F4x15C[NumBoostStates]=1 | | | D18F4x15C[NumBoostStates]=3 | | |
|---|---|---|---|---|---|
| **P-state Name** | **Index Used for Requests/Status** | **Corresponding MSR Address** | **P-state Name** | **Index Used for Requests/Status** | **Corresponding MSR Address** |
| Pb0 | n/a | MSRC001_0064 | Pb0 | n/a | MSRC001_0064 |
| P0 | 0 | MSRC001_0065 | Pb1 | n/a | MSRC001_0065 |
| P1 | 1 | MSRC001_0066 | Pb2 | n/a | MSRC001_0066 |
| P2 | 2 | MSRC001_0067 | P0 | 0 | MSRC001_0067 |
| P3 | 3 | MSRC001_0068 | P1 | 1 | MSRC001_0068 |
| P4 | 4 | MSRC001_0069 | P2 | 2 | MSRC001_0069 |
| P5 | 5 | MSRC001_006A | P3 | 3 | MSRC001_006A |
| P6 | 6 | MSRC001_006B | P4 | 4 | MSRC001_006B |

#### 2.5.3.1.4 Core P-state Visibility

[The P-State Status Register] MSRC001_0063[CurPstate] reflects the number of the current non-boosted P-state of each core. For example, core 1 MSRC001_0063[CurPstate]=010b indicates core 1 is in the P2 state. If a core is in a boosted P-state, MSRC001_0063[CurPstate] reads back as 0.

The voltage being provided to a core may not correspond to the VID code specified by the current P-state of the core due to voltage plane dependencies. See 2.5.2 [Frequency and Voltage Domain Dependencies]. If MSRC001_0063[CurPstate]=0, the frequency of the core could be the frequency specified by P0 or any boosted P-state. To determine the frequency of a core, see 2.5.3.3 [Effective Frequency].

#### 2.5.3.1.5 Core P-state Limits

Core P-states may be limited to lower-performance states under certain conditions, including:

- HTC. See [The Hardware Thermal Control (HTC) Register] D18F3x64[HtcPstateLimit].
- LHTC. See [The Local Hardware Thermal Control (LHTC) Register] D18F3x138[LHtcPstateLimit].
- PROCHOT_L assertion. See 2.10.3.1 [PROCHOT_L and Hardware Thermal Control (HTC)].
- SB-TSI. See SB-TSI_x2F[PstateLmt] in SBI Temperature Sensor Interface (SB-TSI) Specification, #40821.

The current non-boosted core P-state limit is provided in [The P-State Current Limit Register] MSRC001_0061[CurPstateLimit]. In addition, the maximum non-boosted P-state value (lowest-performance P-state), regardless of the source, is limited as specified in MSRC001_0061[PstateMaxVal].

#### 2.5.3.1.6 Core P-state Transition Behavior

The following rules specify how P-states changes function and interact with other system or processor states:

- If the P-state number is increasing (the core is moving to a lower-performance state), then the COF is changed first, followed by the VID change. If the P-state number is decreasing, then the VID is changed first followed by the COF.
- See 2.5.1.5 [Voltage Transitions] for details about voltage transitions made during P-state changes.
- If multiple commands are issued that affect the P-state of a domain prior to when the processor initiates the change of the P-state of that domain, then the processor operates on the last one issued.
- Once a P-state change starts, the P-state state machine (PSSM) continues through completion unless interrupted by a RESET_L deassertion. If multiple P-state changes are requested concurrently, the PSSM may group the associated VID changes separately from the associated COF changes.
- If RESET_L asserts, all cores are transitioned to C0 and to the P-state specified by MSRC001_0071[StartupPstate].
  - After a warm reset [The P-State Control Register] MSRC001_0062 and [The P-State Status Register] MSRC001_0063 are consistent with MSRC001_0071[CurPstate].
- The OS controls the P-state through [The P-State Control Register] MSRC001_0062, independent of the P-state limits described in MSRC001_0061[PstateMaxVal, CurPstateLimit]. P-state limits interact with OS-directed P-state transitions as follows:
  - The P-state for a core is changed by hardware to MSRC001_0061[PstateMaxVal] if MSRC001_0061[PstateMaxVal] changes and the current P-state number is higher than (lower-performance than) MSRC001_0061[PstateMaxVal].
  - The P-state for a core is changed by hardware to MSRC001_0061[CurPstateLimit] if MSRC001_0061[CurPstateLimit] changes and the current P-state number is lower than (higher-performance than) MSRC001_0061[CurPstateLimit].
- See 2.5.3.2.7 [C-state initiated P-state Changes] for information regarding P-state usage while cores are in C-states and after cores exit C-states.

#### 2.5.3.1.7 BIOS Requirements for Core P-State Initialization and Transitions

During initial boot, BIOS performs the following sequence:

1. Check that CPUID Fn8000_0007_EDX[HwPstate]=1. If not, P-states are not supported, no P-state related ACPI objects should be generated, and BIOS must skip the rest of these steps.
2. Complete the requirements in 2.5.1.3 [BIOS Requirements for Power Plane Initialization].
3. Determine the valid set of P-states based on the enabled P-states indicated in [The P-State [7:0] Registers] MSRC001_00[6B:64][PstateEn].
4. Optionally perform 2.5.3.1.8 [Processor-Systemboard Power Delivery Check].
5. If only one P-state is enabled in [The P-State [7:0] Registers] MSRC001_00[6B:64][PstateEn], then BIOS must not generate ACPI-defined P-state objects described in 2.5.3.1.9 [ACPI Processor P-State Objects]. Otherwise, the ACPI objects should be generated to enable P-state support.
6. If D18F4x15C[NumBoostStates]!=0, ensure that the recommended settings have programmed into SMUx0B_x8580 and D18F6x50, then interrupt the SMU with Service Index 12h. See 2.12.1.2 [Software Interrupts].

BIOS also creates an AMD proprietary ACPI method, the ALIB method, to change the following fields during runtime based on whether the system is on battery power (see BatteryPower):

- D18F4x15C[BoostSrc].
- SMUx0B_x8580[PdmEn]:
  - Each time SMUx0B_x8580[PdmEn] is modified, ALIB also interrupts the SMU with Service

Index 12h.

See your AMD representative for details about the ALIB method.

#### 2.5.3.1.8 Processor-Systemboard Power Delivery Check

BIOS may disable core P-states that require higher power delivery than the systemboard can support. This power delivery compatibility check is designed to prevent system failures caused by exceeding the power delivery capability of the systemboard for the VDD power plane. BIOS can optionally notify the user if P-states are detected that exceed the systemboard power delivery capability. Modifications to MSRC001_00[6B:64] must be applied equally to all cores. This check does not guarantee functionality for all package/socket compatible processor/systemboard combinations.

MSRC001_00[6B:64][PstateEn] must be set to 0 for any P-state MSR where PstateEn=1 and the processor current requirement (ProcIddMax), defined by the following equation, is greater than the systemboard current delivery capability.

ProcIddMax = MSRC001_00[6B:64][IddValue] * 1/10^MSRC001_00[6B:64][IddDiv] * (D18F3xE8[CmpCap]+1);

The power delivery check should be applied starting with P0 and continue with increasing P-state indices (1, 2, 3, and 4) for all enabled P-states. Once a compatible P-state is found using the ProcIddMax equation, the check is complete.

Example:
- MSRC001_0065[IddValue] = 32d.
- MSRC001_0065[IddDiv] = 0d.
- D18F3xE8[CmpCap] = 1d.
- ProcIddMax = 32 * 1 * 2 = 64 A.

In this example, the systemboard must be able to supply >= 64 A on VDD in order to support P1 for this processor. If the systemboard current delivery capability is < 64 A per plane then BIOS must clear MSRC001_0065[PstateEn] to 0 for all cores on this processor node, and continue by checking P2 in the same fashion.

If no P-states are disabled on the processor while performing the power delivery compatibility check then BIOS does not need to take any action.

If at least one P-state is disabled on the processor by performing the power delivery compatibility check and at least one P-state remains enabled for the processor, then BIOS must perform the following steps:
1. Clear both D18F4x15C[BoostSrc] and D18F4x15C[NumBoostStates] to 0.
2. If the P-state pointed to by MSRC001_0063[CurPstate] is disabled by the power delivery compatibility check, then BIOS must request a transition to an enabled P-state using MSRC001_0062[PstateCmd] and wait for MSRC001_0063[CurPstate] to reflect the new value.
3. Copy the contents of the enabled P-state MSRs (MSRC001_00[6B:64]) to the highest performance P-state locations and disable any duplicate P-states. E.g. if P0 and P1 are disabled by the power delivery compatibility check and P2 - P4 remain enabled, then the contents of MSRC001_0066 - MSRC001_0068 should be copied to MSRC001_0064 - MSRC001_0066 and MSRC001_00[68:67][PstateEn] should be cleared to 0.
4. Request a P-state transition to the P-state MSR containing the COF/VID values currently applied. E.g. If MSRC001_0063[CurPstate]=100b and P4 P-state MSR information is copied to P2 in step 2, then BIOS

should write 010b to MSRC001_0062[PstateCmd] and wait for MSRC001_0063[CurPstate] to reflect the new value.

5. Adjust the following P-state parameters affected by the P-state MSR copy by subtracting the number of P-states that are disabled by the power delivery compatibility check. This calculation should not wrap, but saturate at 0. E.g. if P0 and P1 are disabled, then each of the following register fields should have 2 subtracted from them:
   - D18F3x64[HtcPstateLimit]
   - D18F3xDC[HwPstateMaxVal]

If the processor has all P-states disabled after performing the power delivery compatibility check, then BIOS must perform the following steps. Note that this does not guarantee operation, and that BIOS should notify the user of the incompatibility between the processor and systemboard if possible.

1. Clear both D18F4x15C[BoostSrc] and D18F4x15C[NumBoostStates] to 0.
2. If MSRC001_0063[CurPstate]!=D18F3xDC[HwPstateMaxVal], then write D18F3xDC[HwPstateMaxVal] to MSRC001_0062[PstateCmd] and wait for MSRC001_0063[CurPstate] to reflect the new value.
3. If D18F3xDC[HwPstateMaxVal]!=000b, copy the contents of the P-state MSR pointed to by D18F3xDC[HwPstateMaxVal] to MSRC001_0064 and set MSRC001_0064[PstateEn]; Write 000b to MSRC001_0062[PstateCmd] and wait for MSRC001_0063[CurPstate] to reflect the new value.
4. Adjust the following fields to 000b.
   - D18F3x64[HtcPstateLimit]
   - D18F3xDC[HwPstateMaxVal]

#### 2.5.3.1.9 ACPI Processor P-State Objects

ACPI 2.0 and ACPI 3.0 processor performance control for processors reporting CPUID Fn8000_0007_EDX[HwPstate]=1 is implemented through two objects whose presence indicates to the OS that the platform and CPU are capable of supporting multiple performance states. Processor performance states are not supported with ACPI 1.0b. BIOS must provide the _PCT object, _PSS object, and define other ACPI parameters to support operating systems that provide native support for processor P-state transitions. Other optional ACPI objects are also described in the following sections.

The following rules apply to BIOS generated ACPI objects in multi-core systems. Refer to the appropriate ACPI specification (http://www.acpi.info) for additional details:

- All cores must expose the same number of performance states to the OS.
- The respective performance states displayed to the OS for each core must have identical performance and power-consumption parameters (e.g. P0 on core 0 must have the same performance and power-consumption parameters as P0 on core1, P1 on core 0 must have the same parameters as P1 on core 1, however P0 can be different than P1)
- Performance state objects must be present under each processor object in the system.

##### 2.5.3.1.9.1 _PCT (Performance Control)

BIOS must declare the performance control object parameters as functional fixed hardware. This definition indicates the processor driver understands the architectural definition of the P-state interface associated with CPUID Fn8000_0007_EDX[HwPstate]=1.

- Perf_Ctrl_Register = Functional Fixed Hardware
- Perf_Status_Register = Functional Fixed Hardware

#### 2.5.3.1.9.2 _PSS (Performance Supported States)

A unique _PSS entry is created for each P-state. The value contained in the _PSS Control field is written to [The P-State Control Register] MSRC001_0062 to request a P-state change to the CoreFreq of the associated _PSS object. The value contained in [The P-State Status Register] MSRC001_0063 can be used to identify the _PSS object of the current P-state by equating MSRC001_0063[CurPstate] to the value of the _PSS Status field. See 2.5.3.1 [Core P-states].

BIOS must loop through each of [The P-State [7:0] Registers] MSRC001_00[6B:64] applying the following formulas to create the fields for the _PSS objects. BIOS must skip over any P-state MSRs that represent boosted P-states (specified by D18F4x15C[NumBoostStates]).

- CoreFreq (MHz) = Calculated using the COF formula documented in MSRC001_00[6B:64][CpuFid].
- Power (mW)
  - Convert MSRC001_00[6B:64][CpuVid] to a voltage by referring to the AMD Voltage Regulator Specification, #40182.
  - Power (mW) = voltage * MSRC001_00[6B:64][IddValue] * 1/10^MSRC001_00[6B:64][IddDiv] * 1000
- TransitionLatency (us) = BusMasterLatency (us) = 0 us.
- Control/Status
  - The highest-performance non-boosted P-state must have the _PSS control and status fields programmed to 0.
  - Each lower-performance non-boosted P-state must have the _PSS control and status fields programmed in ascending order.

For example:
- D18F4x15C[NumBoostStates]=1
- The P-state MSRs have the following values:
  - MSRC001_0064=8000_0020_0000_4800h
  - MSRC001_0065=8000_0023_0000_5840h
  - MSRC001_0066=8000_001C_0000_6800h
  - MSRC001_0067=8000_01BC_0000_7802h
  - MSRC001_00[6B:68]=0

In this example, MSRC001_0064 should be skipped when creating _PSS objects because it specifies a boosted P-state. MSRC001_0065 should be the first MSR referenced when creating _PSS objects. The _PSS objects should have the following values:
- P0
  - CoreFreq=1333MHz
  - Power=35W
  - TransitionLatency=BusMasterLatency=0
  - Control=Status=0
- P1
  - CoreFreq=889MHz
  - Power=25W
  - TransitionLatency=BusMasterLatency=0
  - Control=Status=1
- P2
  - CoreFreq=400MHz
  - Power=15W
  - TransitionLatency=BusMasterLatency=0
  - Control=Status=2

In this example, no other P-states should be defined.

#### 2.5.3.1.9.3 _PPC (Performance Present Capabilities)

The _PPC object is optional. Refer to the ACPI specification for details on use and content.

#### 2.5.3.1.9.4 _PSD (P-State Dependency)

An ACPI 3.0 _PSD object is required for each core:

- NumberOfEntries = 5.
- Revision = 0.
- Domain = 0.
- CoordType = FCh. (SW_ALL)
- NumProcessors = CPUID Fn8000_0008_ECX[NC]+1.

#### 2.5.3.1.9.5 Fixed ACPI Description Table (FADT) Entries

Declare the following FADT entries:

- PSTATE_CNT = 00h.
- DUTY_WIDTH = 00h.

### 2.5.3.1.10 XPSS (Microsoft® Extended PSS) Object

Some Microsoft® operating systems require an XPSS object to make P-state changes function properly. A BIOS that implements an XPSS object has special requirements for the _PCT object. See the Microsoft *Extended PSS ACPI Method Specification* for the detailed requirements to implement these objects.

### 2.5.3.2 C-states

C-states are processor power states. C0 is the operational state in which instructions are executed. All other C-states are low-power states in which instructions are not executed.

#### 2.5.3.2.1 C-state Names and Numbers

C-states are often referred to by an alphanumeric naming convention, C1, C2, C3, etc. The mapping between ACPI defined C-states and AMD specified C-state actions is not direct. The actions taken by the processor when entering a low-power C-state are specified by D18F4x118 and D18F4x11C and are configured by software. See 2.5.3.2.3 [C-state Actions] for information about AMD specific actions.

#### 2.5.3.2.2 C-state Request Interface

C-states are dynamically requested by software and are exposed through ACPI objects (see 2.5.3.2.8 [ACPI Processor C-state Objects]). C-states can be requested on a per-core basis. Software requests a C-state change in one of two ways, either by executing the HLT instruction or by reading from an IO address specified by MSRC001_0073[CstateAddr] plus an offset of 0 through 7 (see D18F4x118 and D18F4x11C for details). Execution of the HLT instruction is equivalent to reading from the IO address specified by MSRC001_0073[CstateAddr]. The processor always returns 0 for this IO read. When software requests a C-state transition:

1. Hardware determines which C-state actions were requested. See 2.5.3.2.3 [C-state Actions].

2. Hardware evaluates any frequency and voltage domain dependencies. See 2.5.2 [Frequency and Voltage Domain Dependencies].
3. Hardware evaluates any enabled and unmasked C-state request monitors. See 2.5.3.2.4 [C-state Request Monitors].
4. Hardware enters the deepest C-state that is allowed.

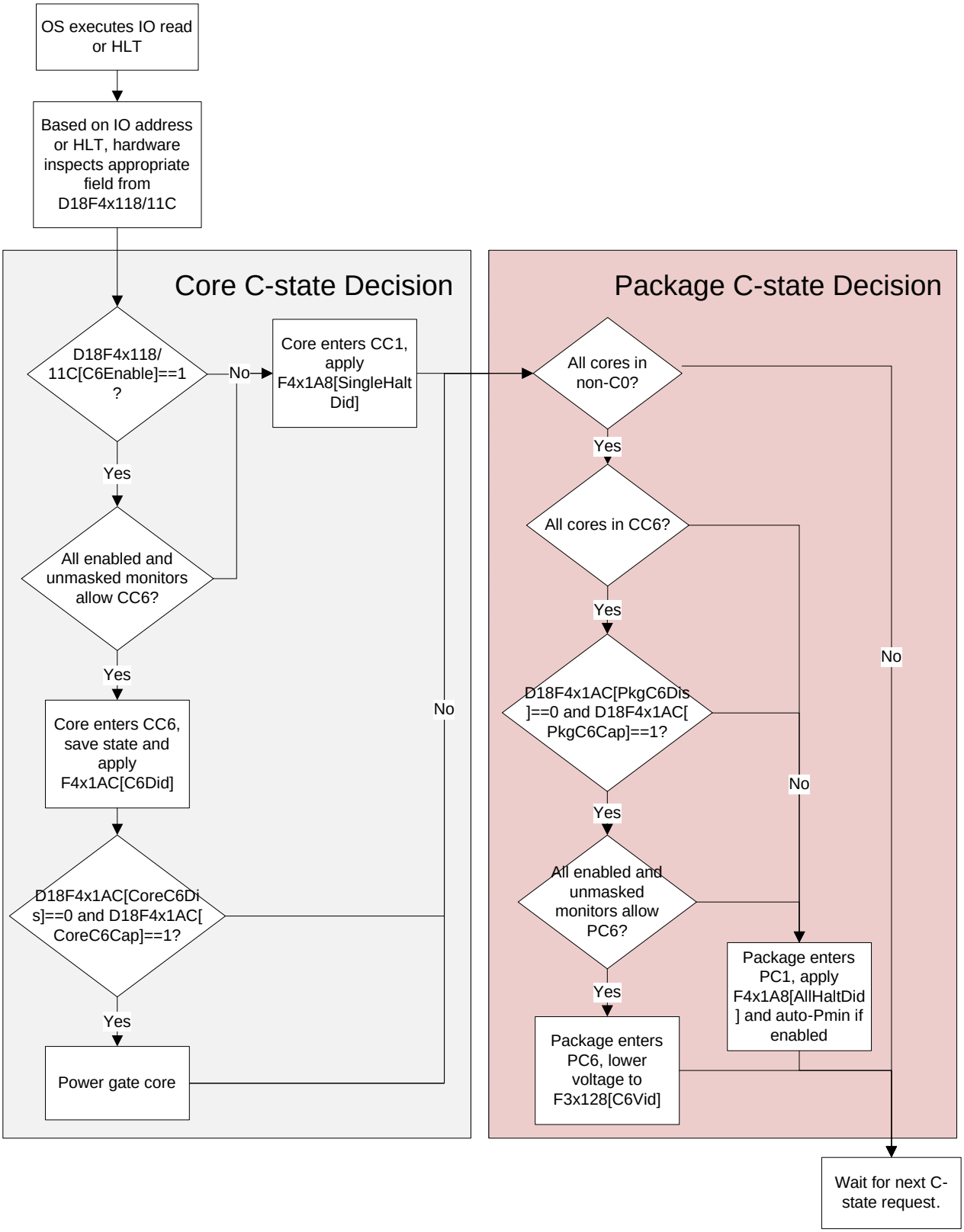


**Figure 3: C-state decision process**

#### 2.5.3.2.3 C-state Actions

Each C-state has per-core and per-package functionality. See 2.5.2.1 [Dependencies Between Cores].

Each C-state action is enabled using D18F4x118 and D18F4x11C. Multiple actions can be enabled for a given IO address. If no actions are specified when a core enters a core C-state, or if hardware determines that no actions are acceptable, the core enters the CC1 state. If no actions are specified when the package enters a package C-state, or if hardware determines that no actions are acceptable, the package enters the PC1 state.

##### 2.5.3.2.3.1 Core C1 (CC1) State

When a core enters the CC1 state, its clock ramps down to the frequency specified by D18F4x1A8[SingleHaltCpuDid].

##### 2.5.3.2.3.2 Core C6 (CC6) State

When a core enters the CC6 state, it executes the following sequence:

1. L1 and L2 caches are flushed to DRAM by hardware.
2. Internal core state is saved to DRAM by hardware.
3. The core clock ramps down to the frequency specified by D18F4x1AC[C6Did].
4. Power is removed from the core if possible as specified by D18F4x1AC[CoreC6Cap] and D18F4x1AC[CoreC6Dis].

The events which cause a core to exit the CC6 state are specified in 2.5.3.2.6 [Exiting C-states].

If a warm reset occurs while a core is in CC6, all MCA registers in the core shown in Table 41 are cleared to 0. See 2.16 [Machine Check Architecture].

###### 2.5.3.2.3.2.1 Core C6 (CC6) Latency Calculations

BIOS should use the following formula for calculating CC6 exit latency:

- 66us + (1 / frequency of lowest performance enabled P-state specified by MSRC001_00[6B:64][PstateEn] in MHz)*13,400

Example calculation assuming MSRC001_00[6B:64][CpuFid, CpuDid] indicate a frequency of 800MHz:

- Exit latency = 66 + (0.00125)*13,400 = 82us

##### 2.5.3.2.3.3 Package C1 (PC1) State

The processor enters the PC1 state with auto-Pmin when all of the following are true:

- All cores are in the CC1 state or deeper.

If D18F4x1AC[CstPminEn] indicates that auto-Pmin is enabled when the processor enters PC1, the P-state for all cores is transitioned as specified by 2.5.3.2.7.1 [Auto-Pmin]. Regardless of the state of D18F4x1AC[CstPminEn], all core clocks are ramped to the frequency specified by D18F4x1A8[AllHaltCpuDid].

##### 2.5.3.2.3.4 Package C6 (PC6) State

The processor enters the PC6 state when all of the following are true:

- All cores enter the CC6 state.
- The C-state action field targeted by each core's C-state request has the C6Enable bit programmed to indicated entry into PC6 is allowed. See D18F4x118 and D18F4x11C.
- PC6 is supported and enabled as specified by D18F4x1AC[PkgC6Cap] and D18F4x1AC[PkgC6Dis].

When the package enters PC6, VDD is transitioned to the VID specified by D18F3x128[C6Vid]

#### 2.5.3.2.4 C-state Request Monitors

Deeper C-states have higher entry and exit latencies but provide greater power savings than shallower C-states. To help balance the performance and power needs of the system, the processor can limit access to specific C-states in certain scenarios.

##### 2.5.3.2.4.1 DMA Tracking

DMA activity can indicate that entering deep C-states may hamper performance. DMA activity tracking allow DMA activity to limit access to certain C-states.

###### 2.5.3.2.4.1.1 DMA Activity Tracking

The processor tracks DMA activity and can disallow access to PC6 as a result. When a package C-state transition request is made, the processor denies access to the respective C-state if any DMA traffic specified by D18F4x120[DeepCstDMATrackEn] has occurred within the hysteresis time specified by D18F4x120[CstDMATrackHyst].

##### 2.5.3.2.4.2 FCH Messaging

The FCH can be notified when the processor transitions package C-states. See D18F4x120[CstateMsgDis]. If the processor sends a message when entering PC6, the FCH sends a return message notifying the processor whether the C-state is allowed. If the C-state is disallowed, the processor enters PC1. The processor can ignore the FCH response as specified by D18F4x120[DeepCstAllowMsgEn].

If the FCH does not respond to the processor within the time specified by D18F4x120[FchTO], a timeout occurs. This causes the processor to take the actions specified by D18F4x120[DeepCstTOPol].

##### 2.5.3.2.4.3 Interrupt Monitors

The processor supports two mechanisms to track interrupt behavior, a timer tick monitor and an interrupt rate monitor. A short timer tick period or a large number of interrupts can indicate that entering deep C-states may hamper performance. Both monitors can control access to CC6 or PC6 independently.

###### 2.5.3.2.4.3.1 Timer Tick Monitor

The timer tick monitor tracks interrupts that are delivered from the FCH on a regular interval, generally referred to as timer tick interrupts. When software changes the period of the timer tick interrupt, the FCH reports the new period to the processor.

The timer tick monitor operates in either interval mode or duration mode as specified by D18F4x124[TrackTimerTickInterEn]. The mode determines the time frame used by the timer tick monitor to make decisions. In interval mode, the monitor uses the timer tick period. In duration mode, the monitor calculates the time remaining until the next expected timer tick interrupt whenever a C-state transition request is made. The monitor compares the time frame specified by the mode to thresholds defined by D18F4x124[IntMonCC6Lmt] and

D18F4x124[IntMonPkgC6Lmt]. If the time frame is less than or equal to the threshold, access to the appropriate C-state is disallowed.

The timer tick monitor is enabled using D18F4x124[IntMonCC6En] and D18F4x124[IntMonPkgC6En].

The timer tick monitor cannot be used to track periodic local APIC timer interrupts.

#### 2.5.3.2.4.3.2 Interrupt Rate Monitor

The interrupt rate monitor tracks the behavior of all interrupts in the system using independent counters for CC6 and PC6. The counters behave as follows:

- On a regular interval specified by D18F4x134[IntRateCC6BurstLen] and D18F4x134[IntRatePkgC6BurstLen], the processor determines if any interrupts were received during the interval. If so, it increments the appropriate counter by 1.
- The processor decrements each counter by 1 at a constant rate, specified by D18F4x134[IntRateCC6DecrRate] and D18F4x134[IntRatePkgC6DecrRate].
- The counters saturate at values specified by D18F4x134[IntRateCC6MaxDepth] and D18F4x134[IntRatePkgC6MaxDepth].

Whenever software requests a C-state transition, the current value in each counter is compared against a threshold specified by D18F4x134[IntRateCC6Threshold] and D18F4x134[IntRatePkgC6Threshold]. If the counter value is greater than the threshold, access to the appropriate C-state is disallowed.

#### 2.5.3.2.4.4 Residency Monitors

Per-core C0 residency and non-C0 residency times are tracked by the processor. Time spent in CC1 can count towards C0 residency time or non-C0 residency time as specified by D18F4x124[MonitorEnableMode]. High C0 residency or low non-C0 residency can indicates system is in use and entering deep C-states may hamper performance. Each residency monitor controls access to CC6 independently.

#### 2.5.3.2.4.4.1 C0 Residency Monitor

The C0 residency monitor uses an internal counter to determine if access to CC6 is allowed. The counter behaves as follows:

- Whenever a core enters a non-C0 C-state, the duration of the core's most recent C0 residency is compared to the threshold specified by D18F4x124[C0MonCC6Lmt]. If the residency was less than the threshold, the counter is incremented, otherwise it is decremented.
- The counter saturates at the value specified by D18F4x124[C0MonCC6Cntr].

When a core attempts to enter CC6, the transition is allowed if the counter is saturated at the value specified by D18F4x124[C0MonCC6Cntr], otherwise access is disallowed.

The C0 residency monitor is enabled using D18F4x124[C0MonCC6En].

#### 2.5.3.2.4.4.2 Non-C0 residency Monitor

The non-C0 residency monitor behaves in one of two ways:

- D18F4x128[NonC0MonCoreOffMode] disables the use of an internal counter: When a core attempts to enter CC6, the transition is allowed if the duration of the core's most recent non-C0 residency was greater than the

threshold specified by D18F4x128[NonC0MonCC6Lmt], otherwise the transition is disallowed.

- D18F4x128[NonC0MonCoreOffMode] enables the use of an internal counter: The internal counter behaves as follows:
  - Whenever a core enters the C0 state, the core's most recent non-C0 residency is compared to the threshold specified by D18F4x128[NonC0MonCC6Lmt]. If the residency was greater than the threshold, the counter is incremented, otherwise it is decremented.
  - The counter saturates at 7.

  When a core attempts to enter CC6, the transition is allowed if the counter is greater than the threshold specified by D18F4x128[NonC0MonCoreOffCntr], others access is disallowed.

The non-C0 residency monitor is enabled using D18F4x128[NonC0MonCC6En].

#### 2.5.3.2.4.5 C-state Monitor Masking

Each of the interrupt and residency C-state request monitors can be masked on a per-CAF basis, as specified by D18F4x1A4. If a C-state monitor is masked for a given CAF, requests to transition to CC6 using that CAF ignore the results of the C-state monitor.

### 2.5.3.2.5 C-states and Probe Requests

If a core is in a C-state in which its caches are not flushed, it must service any probe requests that occur. The following algorithm defines the actions taken by the processor prior to and after servicing probe requests from non-C0 C-states:

```
Probe request comes in to a core in a non-C0 C-state and core caches have not been
flushed;

if (the package is in PC1 && D18F4x1AC[CstPminEn]==1) {
    Ramp the core frequency to D18F4x1A8[PService];
    Service the probe request;
    Wait for D18F3xD4[ClkRampHystSel] timer to expire;
    Ramp the core to the frequency specified by the current C-state;
} elseif (a core is in CC1 while other cores are in C0 || the package is in PC1) {
    If ((D18F4x1A8[CpuProbEn]==0) || (core divisor is /16 or deeper)) {
        Ramp the core to the frequency specified by the current P-state;
        Service the probe request;
        Wait for D18F3xD4[ClkRampHystSel] timer to expire;
        Ramp the core to the frequency specified by the current C-state;
    } else {
        Service the probe request;
    }
}
```

### 2.5.3.2.6 Exiting C-states

The following events may cause the processor to exit a non-C0 C-state and return to C0:

- INTR.
- NMI.
- SMI.
- INIT.
- RESET_L assertion.

If an INTR is received while a core is in a non-C0 C-state, the state of EFLAGS[IF] and the mechanism used to enter the non-C0 C-state determine the actions taken by the processor.

- Entry via HLT, EFLAGS[IF]==0: The interrupt does not wake up the core.
- Entry via HLT, EFLAGS[IF]==1: The interrupt wakes the core and the core begins execution. Any pending interrupts are serviced.
- Entry via IO read, EFLAGS[IF]==0: The interrupt wakes the core and the core begins execution. Any pending interrupts are not serviced until EFLAGS[IF] is programmed to 1.
- Entry via IO read, EFLAGS[IF]==1: The interrupt wakes the core and the core begins execution. Any pending interrupts are serviced.

Regardless of the entry mechanism, if [The Task Priority Register] APIC80 indicates the core is at a higher priority level than the INTR received, the core does not wake up.

#### 2.5.3.2.7 C-state initiated P-state Changes

C-state changes can automatically cause core P-state changes. These features make use of a timer called the PService timer and a P-state called the PService state. The PService timer is enabled using D18F4x1A8[PServiceTmrEn], expires as specified by D18F4x1A8[PServiceTmr], and counts/resets as described below. The PService state is specified by D18F4x1A8[PService].

Any P-state limits in effect are honored when making these P-state changes.

##### 2.5.3.2.7.1 Auto-Pmin

If D18F4x1AC[CstPminEn] indicates auto-Pmin is enabled when the processor enters package C1, hardware automatically transitions all cores to the PService state. In this case, if the PService timer is enabled when the processor exits PC1, the PService timer resets and begins counting down. All cores remain at the PService state until one of the following occurs:

- The PService timer expires while the package is in C0: In this case, the cores transition back to the last P-state requested by software.
- The package enters PC1 with auto-Pmin: In this case, the PService timer stops counting and the cores remain in the PService state. See 2.5.3.2.3.3 [Package C1 (PC1) State].
- The package enters PC6: In this case the PService timer stops counting and the actions associated the requested package C-state occur. See 2.5.3.2.3.4 [Package C6 (PC6) State].

If the PService timer is disabled when the processor exits PC1, the cores transition back to the last P-state requested by software.

##### 2.5.3.2.7.2 Exiting PC6

If the PService timer is enabled when the processor exits PC6, all cores transition to the P-state specified by D18F4x1AC[PstateIdCoreOffExit]. The cores remain in this state until one of the following occurs:

- The PService timer expires while the package is in C0: In this case, the cores transition back to the last P-state requested by software.
- The package enters PC1 without auto-Pmin: In this case, the PService timer continues counting. If it expires while in PC1, the cores remain in the P-state specified by D18F4x1AC[PstateIdCoreOffExit] until they return to C0, at which time they transition to the last P-state requested by software.
- The package enters PC1 with auto-Pmin: In this case, the PService timer stops counting and the cores enter the PService state.

- The package enters PC4 or PC6: In this case the PService timer stops counting and the actions associated the requested package C-state occur. See 2.5.3.2.3.4 [Package C6 (PC6) State].

If the PService timer is disabled when the processor exits PC6, the cores transition back to the last P-state requested by software.

#### 2.5.3.2.8 ACPI Processor C-state Objects

ACPI 2.0 and ACPI 3.0 processor power control is implemented through the _CST object whose presence indicates to the OS that the platform and processor are capable of supporting multiple power states. BIOS must provide the _CST object and define other ACPI parameters to support operating systems that provide native support for processor C-state transitions. Other optional ACPI objects are also described in the following sections.

The _CST object is not supported with ACPI 1.0b. BIOS should provide FADT entries as outlined below to support operating systems that are not ACPI 2.0 capable.

The following rules apply to BIOS generated ACPI objects in multi-core systems. Refer to the appropriate ACPI specification for additional details:

- C-state objects must be present under each processor object in the system.

##### 2.5.3.2.8.1 _CST (C-state)

The _CST objects contains information about each C-state the processor supports. BIOS provides a _CST object for each processor object as follows:

- Count: 1.
- C-state package:
  - Register:
    - AddressSpaceKeyword: SystemIO
    - RegisterBitWidth: 8
    - RegisterBitOffset: 0
    - RegisterAddress: MSRC001_0073[CstateAddr] + 1
    - AccessSize: 1 (byte)
  - Type: 2
  - Latency:
    - PC6enabled = (D18F4x1AC[PkgC6Cap]==1 && D18F4x1AC[PkgC6Dis]==0)
    - CC6enabled = (D18F4x1AC[CoreC6Cap]==1 && D18F4x1AC[CoreC6Dis]==0)
    - If (PC6enabled && CC6enabled) then see 2.5.3.2.3.2.1 [Core C6 (CC6) Latency Calculations] for calculating CC6 exit latency.
    - If (PC6enabled && !CC6enabled) then 100us.
    - If (!PC6enabled) then 5us.
  - Power: 0.

##### 2.5.3.2.8.2 _CRS

BIOS must declare in the root host bridge _CRS object that the IO address range from MSRC001_0073[CstateAddr] to MSRC001_0073[CstateAddr]+7 is consumed by the host bridge.

#### 2.5.3.2.8.3 Fixed ACPI Description Table (FADT) Entries

Declare the following FADT entries:

- P_LVL2_LAT =
  - PC6enabled = (D18F4x1AC[PkgC6Cap]==1 && D18F4x1AC[PkgC6Dis]==0)
  - CC6enabled = (D18F4x1AC[CoreC6Cap]==1 && D18F4x1AC[CoreC6Dis])==0)
  - If (PC6enabled && CC6enabled) then see 2.5.3.2.3.2.1 [Core C6 (CC6) Latency Calculations] for calculating CC6 exit latency.
  - If (PC6enabled && !CC6enabled) then 100us.
  - If (!PC6enabled) then 5us.
- P_LVL3_LAT = 1001us.
- FLAGS.PROC_C1 = 1.
- FLAGS.P_LVL2_UP = 1.

Declare the following P_BLK entries for each core:

- P_LVL2 = MSRC001_0073[CstateAddr] + 1.
- P_LVL3 = 0.

#### 2.5.3.2.9 BIOS Requirements for C-state Initialization

During the initial boot, BIOS performs the following sequence:

1. Initialize MSRC001_0073[CstateAddr] on each core to a region of the IO address map with 8 consecutive available addresses. The cores are not required to use the same IO addresses.
2. Program D18F4x1A8[PService] to the index of lowest-performance P-state with MSRC001_00[6B:64][PstateEn]==1 on core 0.
3. Program D18F4x1AC[CstPminEn] to 1.
4. If CC6 or PC6 is supported as indicated by D18F4x1AC[CoreC6Cap, PkgC6Cap]:
   - Ensure D18F2x118[C6DramLock] and D18F4x12C[C6Base] are programmed as specified by 2.9.6 [DRAM CC6/PC6 Storage].
   - If PC6 is supported, program D18F4x1AC[PstateIdCoreOffExit] to the index of lowest-performance P-state with MSRC001_00[6B:64][PstateEn]==1 on core 0.
   - Program D18F4x118 to 0000_0101h.
5. Generate ACPI objects as described in 2.5.3.2.8 [ACPI Processor C-state Objects].

BIOS also creates an AMD proprietary ACPI method, the ALIB method, to change the following fields during runtime based on whether the system is on battery power (see BatteryPower):
- D18F4x1A4[IntRateMonMask].

See your AMD representative for details about the ALIB method.

### 2.5.3.3 Effective Frequency

The effective frequency interface allows software to discern the average, or effective, frequency of a given core over a configurable window of time. This provides software a measure of actual performance rather than forcing software to assume the current frequency of the core is the frequency of the last P-state requested. This can be useful when the P-state is being limited by HTC, CPB, etc.

The effective frequency can be determined using [The Max Performance Frequency Clock Count (MPERF)]

MSR0000_00E7 and [The Actual Performance Frequency Clock Count (APERF)] MSR0000_00E8 and the following steps:

1. At some point in time, write 0 to both MSRs.
2. At a later point in time, read both MSRs.
3. Effective frequency = (value read from MSR0000_00E8 / value read from MSR0000_00E7) * P0 frequency using software P-state numbering.

Additional notes:
- The amount of time that elapses between steps 1 and 2 is determined by software.
- It is software's responsibility to disable interrupts or any other events that may occur in between the write of MSR0000_00E7 and the write of MSR0000_00E8 in step 1 or between the read of MSR0000_00E7 and the read of MSR0000_00E8 in step 2.
- The behavior of MSR0000_00E7 and MSR0000_00E8 may be modified by MSRC001_0015[EffFreqCntMwait].
- The effective frequency interface provides +/- 50MHz accuracy if the following constraints are met:
  - Effective frequency is read no more often than one time per millisecond.
  - When reading or writing MSR0000_00E7 and MSR0000_00E8 software executes only MOV instructions, and no more than 3 MOV instructions, between the two RDMSR or WRMSR instructions.
- MSRC001_0015[TscFreqSel] must be set to 1 for the effective frequency interface to function correctly.

### 2.5.4 Northbridge Power Management

#### 2.5.4.1 NB P-states

The processor supports dynamic northbridge frequency (NCLK) changes and voltage (VDDNB) change requests, referred to as NB P-states. NB P-states are enabled/disabled with D18F6x90[NbPsCtrlDis]. The northbridge switches between two states, NBP0, a higher-performance higher-power consumption state, and NBP1, a lower-performance lower-power consumption state. The COF and VID controls for each NB P-state are specified by:

- NBP0
  - COF: D18F3xDC[NbPs0NclkDiv].
  - VID: D18F3xDC[NbPs0Vid].
- NBP1
  - COF: D18F6x90[NbPs1NclkDiv].
  - VID: D18F6x90[NbPs1Vid].

No runtime software control is needed for NB P-states. Hardware autonomously transitions the NB P-state when all necessary criteria are met. The following criteria are used by hardware to determine when an NB P-state transitions is necessary:

- Core P-states: Based on the setting of D18F6x94[CpuPstateThrEn], hardware can take the current P-state of each core into account when determining whether to transition NB P-states. If D18F6x94[CpuPstateThrEn]=1 and all cores are in a P-state with equal or lesser performance than the P-state specified by D18F6x94[CpuPstateThr], a transition from NBP0 to NBP1 may be triggered. If D18F6x94[CpuPstateThrEn]=1 and any core is in a greater performance P-state than specified by D18F6x94[CpuPstateThr], a transition from NBP1 to NBP0 may be triggered.
- Core C-states: When all cores have entered a non-C0 C-state and an amount of time specified by D18F6x94[NbPsNonC0Timer] has elapsed, a transition from NBP0 to NBP1 may be triggered.
- GPU activity: As specified by D18F6x90[NbPs1GnbSlowIgn], the GPU driver can specify the level of GPU

activity that can cause an NB P-state transition.
  - If SCLK DPM is enabled (see 2.5.5.1 [Dynamic Power Management (DPM)]), transitions to NBP1 can occur whenever the GPU enters a DPM state with GMMx6E0[SclkDpmGnbSlow]==1. A transition to NBP0 occurs whenever the GPU enters a DPM state with GMMx6E0[SclkDpmGnbSlow]==0.
  - If SCLK DPM is disabled, transitions to NBP1 can occur whenever SCLK is gated and transitions to NBP0 occur whenever SCLK is not gated.
- NB P-state forcing: Software can force an NB P-state transition unconditionally using D18F6x90[NbPsForceSel, NbPsForceReq].
- DRAM self-refresh: If DRAM is in self-refresh, no NB P-state changes occur.

Once hardware determines that an NB P-state transition is necessary, the following parameters may delay the NB P-state transition:

- NB P-state residency timer: When a transition from NBP0 to NBP1 occurs, transitions back to NBP0 are prevented until the time specified by D18F6x94[NbPs1ResTmrMin] has elapsed. When a transition from NBP1 to NBP0 occurs, transitions back to NBP1 are prevented until the time specified by D18F6x94[NbPs0ResTmrMin] has elapsed.
- DMA activity: Based on the setting of D18F6x94[NbPs1NoTransOnDma], DMA activity or the assertion of DMAACTIVE_L can prevent NB P-state transitions.

The following diagrams show how all of the criteria are logically combined together.

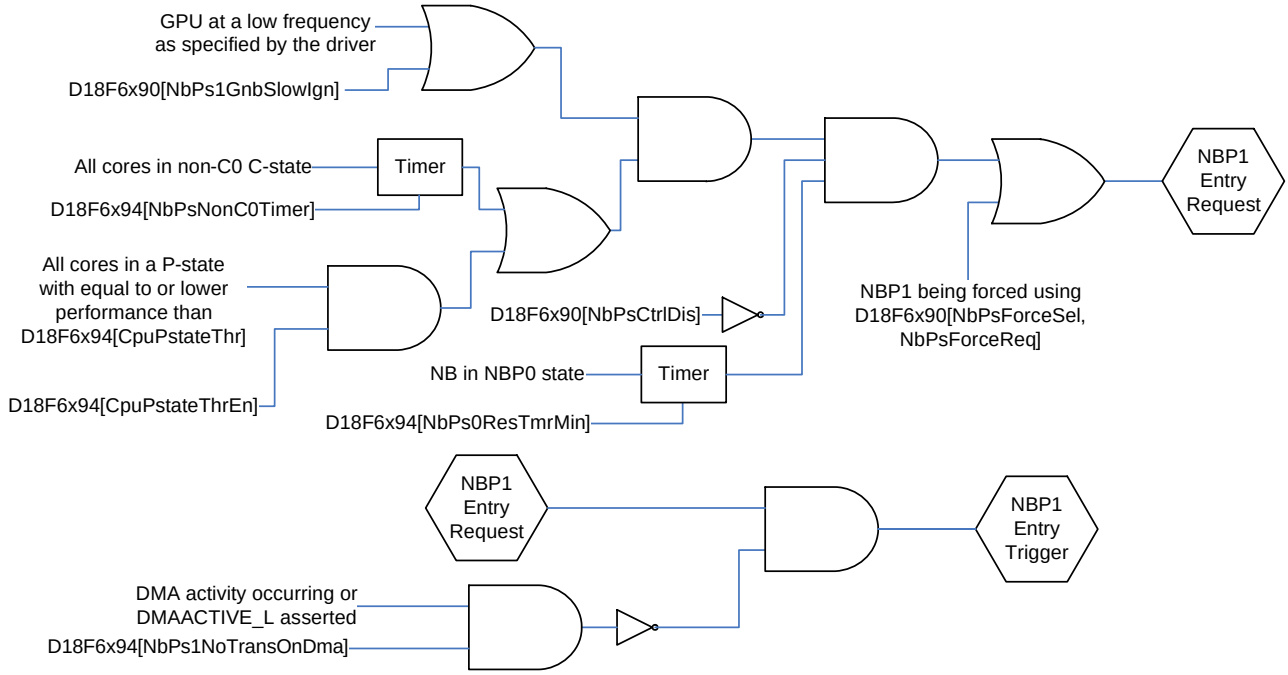


**Figure 4: NBP0 to NBP1 transition determination**

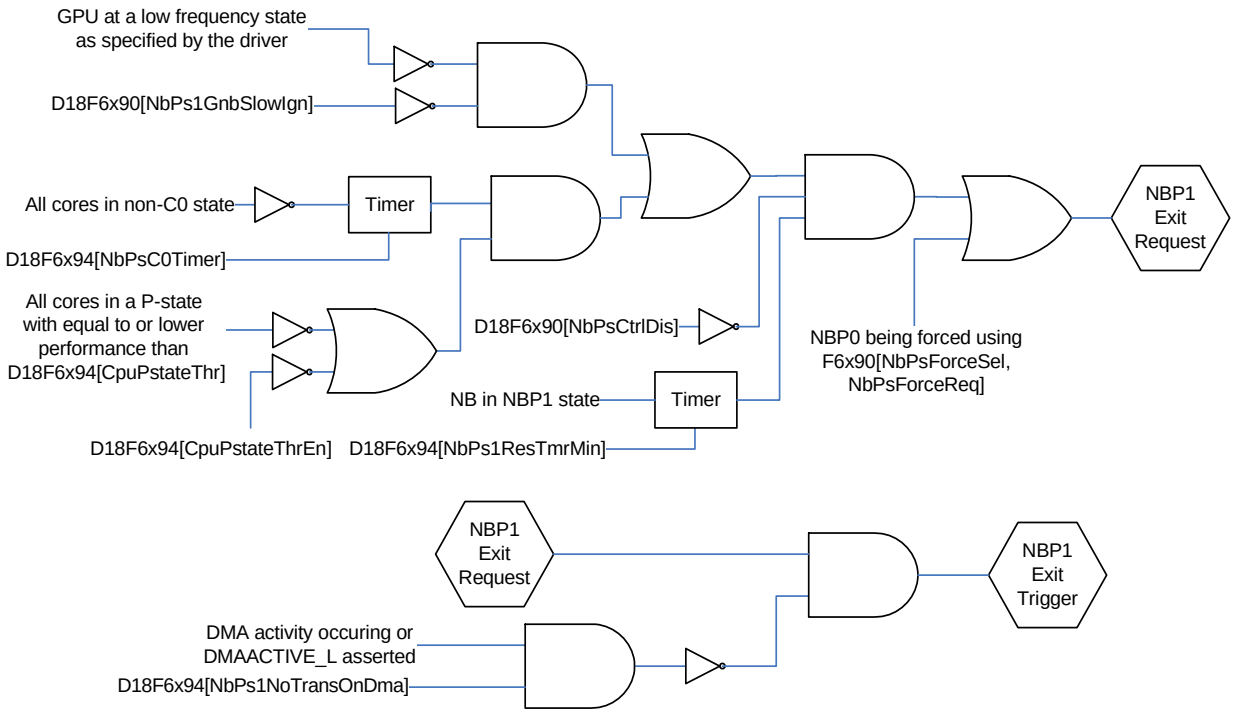


**Figure 5: NBP1 to NBP0 transition determination**

Once hardware determines that a NB P-state transition is necessary and that it may make the transition, hardware executes the following sequence:

1. Set D18F6x98[NbPsTransInFlight] = 1.
2. If the transition is from NBP1 to NBP0, request a VDDNB transition from D18F6x90[NbPs1Vid] to D18F3xDC[NbPs0Vid] and wait for it to complete (see 2.5.1.5 [Voltage Transitions]).
3. Stop memory traffic and place DRAM into self-refresh.
4. Transition from current NCLK divisor to new NCLK divisor as specified by D18F3xDC[NbPs0NclkDiv] and D18F6x90[NbPs1NclkDiv].
5. Update memory settings.
6. Take DRAM out of self-refresh and allow memory traffic.
7. If the transition is from NBP0 to NBP1, request a VDDNB transition from D18F3xDC[NbPs0Vid] to D18F6x90[NbPs1Vid].

When hardware makes an NB P-state change, D18F2x[1,0]F4_x30[DbeGskFifoNumerator] and D18F2x[1,0]F4_x31[DbeGskFifoDenominator] must be programmed to non-zero values or undefined behavior results.

#### 2.5.4.1.1 BIOS Requirements for NB P-state Initialization During DRAM Training

1. Determine the target MEMCLK frequency. See 2.9.3.2.2.1 [Requirements for DRAM Frequency Change During Training].
2. If (target MEMCLK frequency < FCRxFE00_7009[NbPs0NclkDiv] frequency):
   - Program D18F3xDC[NbPs0NclkDiv] to the minimum divisor that generates a 50% duty cycle clock where (target MEMCLK frequency >= (D18F3xD4[MainPllOpFreqId] frequency) / divisor).

   Else:
   - Program D18F3xDC[NbPs0NclkDiv] = FCRxFE00_7009[NbPs0NclkDiv].

3. Wait for D18F3xDC[NclkFreqDone] == 1.
4. If (D18F6x90[NbPsCap]):
   - If (target MEMCLK frequency/2 > FCRxFE00_7006[NbPs1NclkDiv] frequency):
     - Program D18F6x90[NbPs1NclkDiv] to the maximum divisor that generates a 50% duty cycle clock where (target MEMCLK frequency/2 <= (D18F3xD4[MainPllOpFreqId] frequency) / divisor).
     - If (D18F6x90[NbPs1NclkDiv] frequency) <= (FCRxFE00_7006[MaxNbFreqAtMinVid] frequency) program D18F6x90[NbPs1Vid] = FCRxFE00_6002[NbPs1VidAddl].
     - If (D18F6x90[NbPs1NclkDiv] frequency) > (FCRxFE00_7006[MaxNbFreqAtMinVid] frequency) program D18F6x90[NbPs1Vid] = FCRxFE00_6002[NbPs1VidHigh].
   - Else:
     - Program D18F6x90[NbPs1NclkDiv] = FCRxFE00_7006[NbPs1NclkDiv].
     - Program D18F6x90[NbPs1Vid] = FCRxFE00_6000[NbPs1Vid].
5. If ((D18F6x90[NbPsCap]) && (D18F3xDC[NbPs0NclkDiv] frequency < D18F6x90[NbPs1NclkDiv] frequency)):
   - Program D18F6x90[NbPs1NclkDiv] = D18F3xDC[NbPs0NclkDiv].

See 2.9.3.4.7 [NB P-states for DCT/Dram Initialization and Training].

#### 2.5.4.1.2 System BIOS Requirements for NB P-state Operation During POST

In addition to 2.5.4.1.1 [BIOS Requirements for NB P-state Initialization During DRAM Training], BIOS creates a data structure in memory containing information about the processor for use by the GPU driver. Please see your AMD representative for more information.

#### 2.5.4.1.3 Software Controlled NB P-states

Software may use the hardware NB P-state mechanism to force an NB P-state change. This is enabled as specified by D18F6x90[NbPsForceReq]. Once enabled, software selects the current NB P-state by programming D18F6x90[NbPsForceSel]. Whenever software requests an NB P-state transition, it occurs as soon as there are no other NB P-state transitions in flight (D18F6x98[NbPsTransInFlight]) and no DMA activity. The transition occurs regardless of all other criteria listed in 2.5.4.1 [NB P-states]. Software can determine when the transition has completed by reading D18F6x98[NbPs1Act].

### 2.5.4.2 NB Clock Ramping

NCLK is opportunistically ramped down whenever DRAM enters self-refresh and the package is in a C-state as specified by D18F6x9C[NclkRedSelfRefrAlways]. NCLK is ramped down to a divisor specified by D18F6x9C[NclkRedDiv].

When DRAM exits self-refresh NCLK ramps back up to the divisor specified by the current NB P-state (see 2.5.4.1 [NB P-states]). This occurs either serially or in parallel with DDR PHY DLL relock. See D18F6x9C[NclkRampWithDllRelock]

### 2.5.4.3 NB Clock Gating

Portions of NCLK can be gated at certain times. This is enabled using D18F3xDC[NbClockGateEn] and D18F3xD4[DisNclkGatingIdle]. The following describes each portion of NCLK that can be gated and the associated controls.

- IFQ: NCLK distributed to the IFQ is gated when all of the following are true:
  - No traffic through the NB.
  - All cores are in a non-C0 C-state and the IFQ has been empty for the hysteresis time specified by

D18F3xDC[NbClockGateHyst]. See 2.5.3.2.3 [C-state Actions].

- DRAM: NCLK distributed to the DRAM is gated when all of the following are true:
  - DRAM NCLK gating is enabled as specified by D18F3xD4[ClockGatingEnDram].
  - IFQ clock gating is active.
  - DRAM is in self-refresh. See 2.5.6.1 [DRAM Self-Refresh].
  - The display buffer is above the watermark level specified by the current DPM state.

- CNBCIF: NCLK distributed to the CNBCIF of each core is gated when any of the following are true:
  - CNBCIF gating is enabled as specified by D18F3xDC[CnbCifClockGateEn].
  - IFQ gating is active.
  - The core is in CC6. See 2.5.3.2.3 [C-state Actions].

### 2.5.5 Root Complex Power Management

#### 2.5.5.1 Dynamic Power Management (DPM)

The processor supports dynamic SCLK and LCLK frequency changes along with VDDNB voltage change requests, known as Dynamic Power Management (DPM). Once initialized, hardware dynamically monitors processor utilization and adjusts the frequencies and voltage based on that utilization. For both LCLK and SCLK DPM, higher numbered states represent higher performance and lower numbered states represent lower performance. All family 12h processors support DPM.

##### 2.5.5.1.1 Activity Monitor

The processor contains two activity monitors, one for SCLK DPM and one for LCLK DPM. Each activity monitor tracks the usage level of different processor subcomponents. See GMMx690 and SMUx33[BusyCntSel]. A binary signal from each subcomponent is used to determine whether that subcomponent is busy. On each SCLK or LCLK cycle respectively, the activity monitors sample the signal from each unmasked subcomponent. If any given subcomponent reports that it is busy, an accumulator is incremented. For the SCLK DPM monitor, this sampling continues for period specified by the current SCLK DPM state (see 2.5.5.1.2 [SCLK DPM]). For LCLK DPM, this sampling period is specified by SMUx33[SamplingPeriod] and this rate is independent of the rate at which LCLK DPM reads the current value of the activity monitor.

At the end of each sampling period, the current value of the accumulator is placed into a filter and the accumulator is reset. The filter is a 15-stage FIR filter. The filter uses a 15-stage shift register to aggregate the current activity with past activity levels.

Each stage in the filter has two coefficients associated with it, a coefficient for upward trending activity and one for downward trending activity, specified by GMMx6[CC:94] and SMUx[51:35:step2]. Once each filter stage has been updated, the monitor determines which coefficients to use by determining the current trend in activity level. The monitor compares the value in each stage to the previous stage. If the majority of comparisons indicate an increase in activity, the trend is considered to be upwards, otherwise the trend is considered to be downwards.

The final aggregate activity value is calculated using the following equation. After multiplying the value in each stage by the appropriate coefficient, the lower 10-bits of the result are truncated before adding them into the final summation:

aggregate activity value = (Σ(value in each stage* coefficient corresponding to stage and trend))

This aggregate activity value is then used to determine whether the DPM state should be changed. See 2.5.5.1.2 [SCLK DPM] and 2.5.5.1.3 [LCLK DPM].

Additional notes:
- The reset value of each stage of the FIR filter is 0.
- If DPM is disabled and then reenabled, the values in the FIR filters are not reset.

#### 2.5.5.1.2 SCLK DPM

SCLK DPM consists of up to 8 states. Any number of states up through 8 may be used and there is no requirement that the states be contiguous. Each state is made up of the following parameters.

- Valid bit: See GMMx6[88:84][SclkDpmVld].
- Divisor: See GMMx6[88:84][SclkDpmDiv].
- Watermark setting: See GMMx71C[DpmXWm].
- VID: See GMMx788[DpmXVid] and 2.5.2.2 [Dependencies Between Subcomponents on VDDNB].
- GnbSlow: See GMMx6E0[SclkDpmGnbSlow] and 2.5.4.1 [NB P-states].
- Activity thresholds: See GMMx7[48:38,30:28].
- Sampling periods: See GMMx7[6C:50] and 2.5.5.1.1 [Activity Monitor].
- Shallow clock divisor: GMMx830[FstateSsDiv] and 2.5.5.2 [LCLK Clock Ramping (Deep Sleep)].
- Deep clock divisor: GMMx724[FstateDsDiv] and 2.5.5.2 [LCLK Clock Ramping (Deep Sleep)].

When SCLK DPM is first enabled, the state is set as specified by GMMx720[DpmStartState] and the activity monitor begins calculating activity levels (see 2.5.5.1.1 [Activity Monitor]). Each time a new aggregate activity value is calculated, it is compared to the activity thresholds of the current SCLK DPM state. If the aggregate activity value is greater than or equal to GMMx7[48:38,30:28][Raising], the state is changed to the next higher numbered (higher performance) valid state. If the aggregate activity value is less than or equal to GMMx7[48:38,30:28][Lowering], the state is changed to the next lower numbered (lower performance) valid state.

After a SCLK DPM state change occurs, the SCLK DPM activity monitor continues to measure activity and add new activity values into its FIR filter at the specified sample rate. However, no aggregate activity values are calculated or compared against until 16 sampling periods have passed.

SCLK DPM is enabled or disabled with GMMx6E0[DpmEn]. Voltage changes as a result of DPM state transitions are enabled or disabled with GMMx788[DpmVidChangeEn].

During runtime, SCLK DPM parameters are programmed by the graphics driver. To enable the driver, BIOS creates a data structure in memory containing information about the processor. Please see your AMD representative for more information.

#### 2.5.5.1.3 LCLK DPM

LCLK DPM operates in one of two modes, either by tracking root complex activity, or by tracking PCIe bandwidth and GFX DMA activity. This is specified by SMUx0B_x8434[LclkDpmType].

LCLK DPM is enabled by setting SMUx0B_x8434[LclkDpmEn] to 1 and interrupting the SMU with Service Index 08h. See 2.12.1.2 [Software Interrupts]. Voltage changes due to LCLK DPM state changes are enabled using FCRxFF30_01E4[VoltageChangeEn].

#### 2.5.5.1.3.1 LCLK DPM Using Root Complex Activity

LCLK DPM using root complex activity consists of up to 8 states. Any number of states up through 8 may be used and there is no requirement that the states be contiguous. Each state is made up of the following parameters:

- Valid bit: SMUx0B_x8490[LclkStateXValid].
- Divisor: SMUx0B_x84[8C:88:step4].
- VID: SMUx0B_x84[9C:94:step4][BaseVid].
- Activity thresholds: SMUx0B_x84[7C:60].
- Sampling periods: SMUx0B_x84[48:38:step4].

When LCLK DPM using root complex activity is enabled, the aggregate activity value from the activity monitor is read as specified by the sampling period of the current state. Each time the aggregate activity is read, it is compared to the activity thresholds of the current DPM state. If the aggregate activity is greater than SMUx0B_x84[7C:60][Raising], the DPM state is changed to the next higher numbered (higher performance) valid state. If the aggregate activity value is less than SMUx0B_x84[7C:60][Lowering], the DPM state is changed to the next lower numbered (lower performance) valid state.

#### 2.5.5.1.3.2 LCLK DPM Using PCIe Bandwidth and GFX DMA Activity

LCLK DPM using PCIe bandwidth and GFX DMA activity consists of up to 10 PCIe bandwidth states and 8 GFX DMA activity states. Any number of states may be used and there is no requirement that the states be contiguous. The PCIe bandwidth states and the GFX DMA activity states transition independently from each other.

Each PCIe bandwidth state is made up of the following parameters:

- Base divisor: SMUx0B_x84[E0:D8].
- Base VID: SMUx0B_x84[9C:94:step4][BaseVid].
- Base VID frequency tolerance: SMUx0B_x84[D4:D0].
- Tolerance exceeded VID: SMUx0B_x84[9C:94:step4][TolExcdVid].
- PCIe bandwidth thresholds: SMUx[5D:55:step2].
- Up hysteresis: SMUx0B_x84[54:4C].
- Down hysteresis: SMUx0B_x84[C8:C4].
- Sampling period: SMUx0B_x84[48:38:step4].

Each GFX DMA activity state is made up of the following parameters:

- Valid bit: SMUx0B_x8490[LclkStateXValid].
- Divisor offset: SMUx0B_x84[8C:88:step4].
- GFX DMA activity thresholds: SMUx0B_x84[7C:60].

When LCLK DPM is enabled using PCIe bandwidth and GFX DMA activity, the PCIe lane counters and activity monitor are sampled as specified by the sampling period of the current PCIe bandwidth state.

Each time the sampling period expires, the PCIe bandwidth specified by D0F0xE4_x013[1:0]_8031[LnCntBandwidth] is compared against the thresholds in SMUx[5D:55:step2]. Hardware determines which two thresholds the current bandwidth falls between and targets the specified PCIe bandwidth state. Two internal counters are used to provide hysteresis. If the target state is higher performance than the current state, the first counter is incremented. If the target state is lower performance than the current state, the second counter is

incremented. The PCIe bandwidth state is transitioned to the target state if the first counter equals SMUx0B_x84[54:4C][FstateUpHyst] or the second counter equals SMUx0B_x84[C8:C4][FstateDnHyst], otherwise the PCIe bandwidth state is not transitioned. When a state transition is made, both counters are reset to 0.

Once a PCIe bandwidth state is determined, the activity monitor is sampled and the GFX DMA activity is compared to the activity thresholds of the current GFX DMA activity state. If the activity is greater than SMUx0B_x84[7C:60][Raising], the GFX DMA activity state is raised by one. If the activity is less than SMUx0B_x84[7C:60][Lowering], the state is lowered by one.

Hardware calculates the final LCLK divisor by subtracting SMUx0B_x84[8C:88:step4][FstateDiv] of the GFX DMA activity state from SMUx0B_x84[E0:D8][BaseDiv] of the PCIe bandwidth state.

Hardware determines the VDDNB voltage request by comparing the final LCLK divisor to SMUx0B_x84[D4:D0][FstateDivTol] of the current PCIe bandwidth state. If the final LCLK divisor is less than or equal to SMUx0B_x84[D4:D0][FstateDivTol], SMUx0B_x84[9C:94:step4][BaseVid] is requested. Otherwise, SMUx0B_x84[9C:94:step4][TolExcdVid] is requested.

#### 2.5.5.1.3.3 BIOS Initialization for LCLK DPM Using Root Complex Activity

1. Ensure the following fields are programmed to their BIOS recommendations:
   - D0F0x64_x6A[VoltageChangeEn].
   - FCRxFF30_01E4[VoltageChangeEn].
   - SMUx0B_x8434[LclkTimerPeriod, LclkTimerPrescalar, LclkDpmEn].
   - SMUx[51:35:step2][UpTrendCoef, DownTrendCoef].
2. Program the following fields:
   - SMUx33[BusyCntSel] = 3.
   - SMUx0B_x8434[LclkDpmType] = 1.
3. Use the following algorithm to program the valid bit, divisor, and VID for each state:

```
int CurState = 7;
for (i = 3; i >= 0; i--) {
   if (FCRxFE00_7113[LclkDpmValid[i]] == 1) {
      Program SMUx0B_x8490[LclkState[CurState]Valid] = 1;
      Program SMUx0B_x84[9C:94:step4][BaseVid[CurState]] = i;
      if (i == 3) {
         Program SMUx0B_x84[8C:88:step4][FstateDiv[CurState]] =
                                          FCRxFE00_7113[LclkDpmDid3];
      } else {
          Program SMUx0B_x84[8C:88:step4][FstateDiv[CurState]] =
                                          FCRxFE00_7110[LclkDpmDid[i]];
      } // endif
      CurState--;
   } // endif
} // endfor
```

4. For each valid state found in step 3, program the following:
   - SMUx0B_x84[7C:60][Lowering] = 10.2 * (SMUx0B_x84[8C:88:step4][FstateDiv] frequency in MHz) - 1.0
   - SMUx0B_x84[7C:60][Raising] = 40.7* (SMUx0B_x84[8C:88:step4][FstateDiv] frequency in MHz) + 99.0
   - SMUx0B_x84[48:38:step4][FstatePeriod] = C350h.

5. Program SMUx33[LclkActMonUnt, LclkActMonPrd] such that the activity monitor sampling period = 65535 * (REFCLK frequency in Hz) / SMUx0B_x84[8C:88:step4][FstateDiv] frequency in Hz for LCLK DPM state 7.
6. Interrupt the SMU with Service Index 08h. See 2.12.1.2 [Software Interrupts].

#### 2.5.5.2 LCLK Clock Ramping (Deep Sleep)

LCLK can be ramped down when various subcomponents of the root complex are idle. This is known as deep sleep.

##### 2.5.5.2.1 LCLK Deep Sleep

When the subcomponents specified by SMUx1B[LclkDpSlpMask] have been idle for the amount of time specified by SMUx1D[LclkDpSlpHyst], LCLK is ramped down to the divisor specified by SMUx1B[LclkDpSlpDiv]. Whenever any of the subcomponents SMUx1B[LclkDpSlpMask], LCLK is ramped up to the divisor specified by the current LCLK DPM state (see 2.5.5.1.3 [LCLK DPM]).

LCLK ramping is enabled or disabled using SMUx1B[LclkDpSlpEn]. LCLK ramping is enabled by BIOS.

###### 2.5.5.2.1.1 BIOS Initialization for LCLK Deep Sleep

1. Ensure the following registers are programmed to their BIOS recommendations:
   - SMUx1B[LclkDpSlpMask, LclkDpSlpDiv].
   - SMUx1D[LclkDpSlpHyst].
2. Program SMUx1B[LclkDpSlpEn] = 1.

#### 2.5.5.3 GPU and Root Complex Clock Gating

To save power, several subcomponents of the GPU and the root complex can gate clock branches when logic is idle. The following is a list of registers that software configures to enable clock gating for each subcomponent.

Display:
- GMMx4D0[SymclkbGateDisable, SymclkaGateDisable, SclkGateDisable, DispclkRDccgGateDisable, DispclkDccgGateDisable].
- FCRxFF30_01F4[CgDcCgttSclkOverride].
- FCRxFF30_01F5[CgDcCgttDispclkOverride].
- GMMx[79,6D]F0[CrtcDispclkGSclGateDisable, CrtcDispclkGDcpGateDisable, CrtcDispclkRDcfeGateDisable]

GPU PCI Interface:
- D0F0x64_x24
- FCRxFF30_01F4[CgBifCgttSclkOverride].
- FCRxFF30_1512
- FCRxFF30_1529

Graphics memory controller:
- FCRxFF30_01F4[CgMcdwCgttSclkOverride, CgMcbCgttSclkOverride].
- FCRxFF30_01F5[CgVmcCgttSclkOverride].
- GMMx15C0
- GMMx20[C0:B8]
- GMMx2478
- GMMx26[58:50]

Root complex:
- D0F0x64_x22
- D0F0x64_x23
- FCRxFF30_01F5[CgIocCgttLclkOverride, CgIocCgttSclkOverride].
- D0F0x98_x4[A:9]
- D0F0x98_x4B
- FCRxFF30_01F5[CgOrbCgttLclkOverride, CgOrbCgttSclkOverride].
- D0F0xE4_x0[2:1]01_0011[DynClkLatency].
- D0F0xE4_x013[2:0]_8011[TxclkLcntEnable, TxclkPermGateLatency, TxclkRegsGateLatency, TxclkRegsGateEnable, TxclkPermStop, TxclkDynGateEnable, TxclkPermGateEven, TxclkDynGateLatency].
- D0F0xE4_x013[2:0]_8012.
- D0F0xE4_x013[2:0]_8014.
- D0F0xE4_x013[2:0]_8015[RefclkBphyGateEnable, RefclkBphyGateLatency, RefclkRegsGateEnable, RefclkRegsGateLatency].
- D0F0xE4_x013[2:0]_8016[LclkDynGateEnable, LclkGateFree, LclkDynGateLatency].

SMU:
- SMUx6F
- SMUx71
- SMUx73

#### 2.5.5.4 GPU and Root Complex Power Gating

The processor supports power gating of the GPU, GMC, UVD and root complex subcomponents. The behavior and initialization of this functionality is dependent on system configuration as specified in the following sections.

##### 2.5.5.4.1 GPU Power Gating

- When the internal GPU is enabled (see GpuEnabled and D0F0x7C[ForceIntGfxDisable]), hardware gates power to the GPU when it is idle. Hardware saves the GPU's state internally.
- When the internal GPU is disabled by BIOS, BIOS is responsible for power gating the GPU.

###### 2.5.5.4.1.1 Software Requirements for GPU Power Gating

When the internal GPU is enabled (see GpuEnabled and D0F0x7C[ForceIntGfxDisable]), GPU power gating is configured by the driver. When the internal GPU is present but disabled by BIOS, BIOS performs the following steps to power gate the GPU:

1. Program the following registers:
   - SMUx0B_x8600 = 09865000h.
   - SMUx0B_x8604 = 0070FE14h.
   - SMUx0B_x8608 = C0090700h.
   - SMUx0B_x860C = 70FE04C0h.
   - SMUx0B_x8610 = 09060070h.
   - SMUx0B_x8614 = FE04C001h.
   - SMUx0B_x8618 = 0000A0FEh.
   - SMUx0B_x861C = 14C00907h.
   - SMUx0B_x8620 = 00A0FE04h.

- SMUx0B_x8624 = C0090600h.
- SMUx0B_x8628 = A0FE04C0h.
- SMUx0B_x862C = 010000B0h.
- SMUx0B_x8630 = FE14C009h.
- SMUx0B_x8634 = 0700B0FEh.
- SMUx0B_x8638 = 04C00906h.
- SMUx0B_x863C = 00B0F004h.
- SMUx0B_x8640 = C0010000h.

2. Program the following registers (see SMUx0B_x86[A0:50:step4]):
   - SMUx0B_x8650 = 76543210h.
   - SMUx0B_x8654 = FEDCBA98h.
   - SMUx0B_x8658 = 00000080h.
   - SMUx0B_x865C = 00320032h.
   - SMUx0B_x8660 = 00100010h.
   - SMUx0B_x8664 = 00320032h.
   - SMUx0B_x8668 = 00000000h.
   - SMUx0B_x866C = 76543210h.
   - SMUx0B_x8670 = FEDCBA98h.
   - SMUx0B_x8674 = 00000080h.
   - SMUx0B_x8678 = 00320032h.
   - SMUx0B_x867C = 00100010h.
   - SMUx0B_x8680 = 00320032h.
   - SMUx0B_x8684 = 00000000h.
   - SMUx0B_x8688 = 76543210h.
   - SMUx0B_x868C = FEDCBA98h.
   - SMUx0B_x8690 = 00000080h.
   - SMUx0B_x8694 = 00320032h.
   - SMUx0B_x8698 = 00100010h.
   - SMUx0B_x869C = 00320032h.
   - SMUx0B_x86A0 = 00000000h.
3. Power gate the GPU by interrupting the SMU with Service Index 0Bh. See 2.12.1.2 [Software Interrupts].

#### 2.5.5.4.2 GMC Power Gating

- When the internal GPU is enabled (see GpuEnabled and D0F0x7C[ForceIntGfxDisable]), hardware gates power to the GMC when all of the following are true:
  - All CPU cores are in a non-C0 C-state.
  - No pending traffic from the link.
  - The GMC is idle.

  Once all requirements are met, hardware saves the GMC's state internally (see 2.14.2.1.1 [Register Save/Restore Engine (RENG)]) and the GMC is power gated. GMC power gating is enabled by BIOS.
- When the internal GPU is disabled by BIOS, BIOS is responsible for power gating the GMC.

##### 2.5.5.4.2.1 BIOS Initialization for GMC Power Gating

When the internal GPU is enabled (see GpuEnabled and D0F0x7C[ForceIntGfxDisable]), BIOS initializes GMC power gating using the following steps:

1. Ensure the GMC is initialized. See 2.14.2.1 [GMC Initialization].
2. Program the following registers:
   - SMUx0B_x8408 = 00000000h.

- SMUx0B_x840C = 00000000h.
- SMUx0B_x8410 = 00000009h.
- SMUx0B_x84A0 = 00320099h.
- SMUx0B_x84A4 = 00320000h.
- SMUx0B_x84A8 = 000A000Ah.

3. Initialize GMC power gating by interrupting the SMU with Service Index 01h. See 2.12.1.2 [Software Interrupts].

When the internal GPU is disabled by BIOS, BIOS performs the following steps to power gate the GMC:

1. Program the following registers:
   - SMUx0B_x8600 = 03865000h.
   - SMUx0B_x8604 = 0060FE14h.
   - SMUx0B_x8608 = C0090700h.
   - SMUx0B_x860C = 60FE04C0h.
   - SMUx0B_x8610 = 09060060h.
   - SMUx0B_x8614 = FE04C001h.
2. Program the following registers (see SMUx0B_x86[A0:50:step4]):
   - SMUx0B_x8650 = 76543210h.
   - SMUx0B_x8654 = FEDCBA98h.
   - SMUx0B_x8658 = 00000008h.
   - SMUx0B_x865C = 00320032h.
   - SMUx0B_x8660 = 00100010h.
   - SMUx0B_x8664 = 00320032h.
   - SMUx0B_x8668 = 00000000h.
3. Power gate GMC by interrupting the SMU with Service Index 0Bh. See 2.12.1.2 [Software Interrupts].

#### 2.5.5.4.3 UVD Power Gating

UVD power gating is controlled by software and is executed when UVD is not in use. UVD's internal state is not saved and it goes through an internal reset when power is restored.

##### 2.5.5.4.3.1 Software Requirements for UVD Power Gating

When the internal GPU is enabled (see GpuEnabled and D0F0x7C[ForceIntGfxDisable]), UVD power gating is initialized and controlled by the driver.

When the internal GPU is disabled by BIOS, BIOS performs the following steps to power gate UVD:

1. Program the following registers:
   - SMUx0B_x8408 = 00000000h.
   - SMUx0B_x840C = 00000000h.
   - SMUx0B_x8410 = 20000008h.
   - SMUx0B_x84A0 = 00320071h.
   - SMUx0B_x84A4 = 00320032h.
   - SMUx0B_x84A8 = 00320032h.
2. Initialize UVD power gating by interrupting the SMU with Service Index 01h. See 2.12.1.2 [Software Interrupts].
3. Power gate UVD by interrupting the SMU with Service Index 03h. See 2.12.1.2 [Software Interrupts].

#### 2.5.5.4.4 Gfx Link Core Power Gating

The Gfx link core can be power gated by software when not in use. If an IO device is connected to the core, or if any of the lanes form a PCIe link that supports hot-plug, the core may not be power gated. If any lanes are being used as a DDI interface, the core may be powered off when the DDI interface is idle. The core may also be powered on to allow access to a PLL within the core and powered off when the register access is complete.

Since the core and each lane can be configured in a variety of ways, several software and hardware components are involved in determining whether to initiate power gating. During boot, BIOS determines the initial configuration of all PCIe lanes. During runtime, the display driver determines whether any DDI links are active or whether it needs access to a PLL within the Gfx link core. Lastly, during runtime, ACPI methods determine whether any hot-pluggable PCIe links have become active or inactive. Given the inputs from BIOS, the driver, and ACPI methods, hardware dynamically powers the Gfx link core when on or off as appropriate.

##### 2.5.5.4.4.1 Software Initialization for Gfx Link Core Power Gating

BIOS must execute steps 1 and 2 during boot to initialize power gating functionality on the Gfx link core.

1. Program the following registers:
   - SMUx0B_x8408 = 00000000h.
   - SMUx0B_x840C = 00000000h.
   - SMUx0B_x8410 = 30000009h.
   - SMUx0B_x84A0 = 00320032h.
   - SMUx0B_x84A4 = 00320032h.
   - SMUx0B_x84A8 = 00320032h.
2. Initialize gfx link core power gating by interrupting the SMU with Service Index 01h. See 2.12.1.2 [Software Interrupts].
3. After performing the link initialization sequence in 2.11.4.2 [Link Configuration and Initialization]:
   - For all Gfx-link PCIe lanes that are connected to an IO device, form a link that supports hot-plug, or are allocated as DDI:
     - Configure SMUx0B_x858C[UpperLaneId, LowerLaneId] with the appropriate lane numbers and SMUx0B_x858C[Core, Tx, Rx]=[1, 0, 0].
     - Interrupt the SMU with Service Index 14h. See 2.12.1.2 [Software Interrupts].
   - If (D1F0x00!=FFFF_FFFFh), for all lanes allocated as DDI:
     - Configure SMUx0B_x858C[UpperLaneId, LowerLaneId] with the appropriate lane numbers and SMUx0B_x858C[Core, Tx, Rx]=[0, 0, 1].
     - Interrupt the SMU with Service Index 13h. See 2.12.1.2 [Software Interrupts].
     - Configure SMUx0B_x859C[PllId] with the appropriate PLL index.
     - Interrupt the SMU with Service Index 18h. See 2.12.1.2 [Software Interrupts].
   - If (D1F0x00==FFFF_FFFFh), for all lanes allocated as DDI:
     - Configure SMUx0B_x858C[UpperLaneId, LowerLaneId] with the appropriate lane numbers and SMUx0B_x858C[Core, Tx, Rx]=[1, 1, 1].
     - Interrupt the SMU with Service Index 13h. See 2.12.1.2 [Software Interrupts].
   - For all PCIe lanes that support hot-plug but are unpopulated:
     - Configure SMUx0B_x858C[UpperLaneId, LowerLaneId] with the appropriate lane numbers and SMUx0B_x858C[Core, Tx, Rx]=[0, 1, 1].
     - Interrupt the SMU with Service Index 13h. See 2.12.1.2 [Software Interrupts].
   - For all unallocated lanes and all unpopulated lanes that do not support hot-plug:
     - Configure SMUx0B_x858C[UpperLaneId, LowerLaneId] with the appropriate lane numbers and SMUx0B_x858C[Core, Tx, Rx]=[1, 1, 1].
     - Interrupt the SMU with Service Index 13h. See 2.12.1.2 [Software Interrupts].

#### 2.5.5.4.4.1.1 ACPI Method Requirements

In order to support Gfx-link power gating, the BIOS must generate ACPI methods and interrupt handlers to detect hot-plug and hot-unplug events. The exact implementation and location of these ACPI methods is system specific, but methods must follow the below guidelines:

- When a hot-plug event occurs:
  - Configure SMUx0B_x858C[UpperLaneId, LowerLaneId] with the appropriate lane numbers and SMUx0B_x858C[Core, Tx, Rx]=[0, 1, 1].
  - Interrupt the SMU with Service Index 14h. See 2.12.1.2 [Software Interrupts].
- When a hot-unplug event occurs:
  - Configure SMUx0B_x858C[UpperLaneId, LowerLaneId] with the appropriate lane numbers and SMUx0B_x858C[Core, Tx, Rx]=[0, 1, 1].
  - Interrupt the SMU with Service Index 13h. See 2.12.1.2 [Software Interrupts].

### 2.5.5.5 PCIe Speed Power Policy (PSPP)

The processor supports dynamically changing the link frequency due to changes in system configuration and power policy, referred to as PCIe Speed Power Policy (PSPP).

#### 2.5.5.5.1 ALIB

PSPP uses both driver calls and interrupts to call an AMD proprietary ACPI method, the ALIB method. See your AMD representative for details about the ALIB method. Several functions are provided by the ALIB method, including the ability for device drivers to enter themselves into a registry of devices maintained by the method. Registered and unregistered devices can be treated differently by some power policies.

Please contact your operating system vendor to determine how drivers call ACPI methods.

#### 2.5.5.5.2 PSPP Power Policies

PSPP relies on predefined system power policies to determine the maximum PCIe frequency at any given time based on system configuration. The policy is chosen by BIOS during boot time. The following policies are recommended by AMD:

**Table 7: PSPP Maximum PCIe Speed**

| Policy | Registered Device When Docked | Unregistered Device When Docked | Registered Device When Undocked | Unregistered Device When Undocked |
|---|---|---|---|---|
| Performance | Gen2 | Gen2 | Gen2 | Gen2 |
| Balanced-High | Gen2 | Gen2 | Gen2 | Gen1 |
| Balanced-Low | Gen2 | Gen1 | Gen2 | Gen1 |
| Power Savings | Gen1 | Gen1 | Gen1 | Gen1 |

Regardless of the power policy, software always configures UMI to run at gen1.

During POST, BIOS enforces the power policy. BIOS requests the appropriate PCIe frequency when initializing each PCIe device. To enable unregistered devices to switch between gen2 and gen1 frequencies when docking and undocking, when using the Balanced-High power policy BIOS boots all devices to then gen2 frequency first, then transitions unregistered devices to gen1 if necessary.

During runtime, the power policies are enforced by the ALIB method. ACPI SCI interrupt handlers call the ALIB method to notify PSPP of docking or undocking events. In response, the ALIB method transitions each device in the system to the appropriate frequency.

### 2.5.6 DRAM Power Management

#### 2.5.6.1 DRAM Self-Refresh

DRAM is placed into self-refresh in the following three scenarios:
- While in S3 as specified by the DramSr bit in D18F3x80 and D18F3x84.
- While in S0, due to NB P-state transitions (see 2.5.4.1 [NB P-states]).
- While in S0, due to stutter mode (see 2.5.6.1.1 [Stutter Mode]).

##### 2.5.6.1.1 Stutter Mode

DRAM is most commonly placed in self-refresh due to stutter mode. The display buffer in the GPU is a combination of a large buffer known as the DMIF (Display Memory Interface FIFO) and a smaller line buffer. The DMIF takes data originating from DRAM, and sends it to the line buffer to draw to the screen. When the data level in the DMIF is full, DRAM is placed in self-refresh, and incoming DRAM requests are queued. As the DMIF drains, it eventually falls below a predefined watermark level, at which point hardware pulls DRAM out of self-refresh and services all the requests in the queue. Once all the requests are complete and the DMIF is full again, a transition back into self-refresh occurs if the stutter mode conditions are still met.

The following requirements must be met before hardware places DRAM into self-refresh:
- D18F4x1A8[DramSrEn]==1.
- All cores are in a non-C0 C-state.
- One of the following is true:
  - The GPU is idle and the internal display buffer is full.
  - The internal GPU is disabled.
- No pending traffic from the link.

Once the above requirements are met:
- If D18F4x1A8[DramSrHystEnable]==0, hardware places DRAM into self-refresh immediately.
- If D18F4x1A8[DramSrHystEnable]==1, hardware waits an amount of time specified by D18F4x1A8[DramSrHyst]. If any of the above requirements are violated during that time, hardware aborts the attempt to enter self-refresh, resets the timer, and attempts to enter self-refresh when the requirements are met once again. If the time specified by D18F4x1A8[DramSrHyst] expires without any of the above requirements being violated, hardware places DRAM into self-refresh.

Once DRAM is in self-refresh, hardware removes it from self-refresh whenever any of the above requirements are not longer met.

To save additional power, hardware can tristate MEMCLK and shut down the DDR phy DLL when DRAM is placed in self-refresh. See D18F4x1A8[MemTriStateEn]. If D18F2x[1,0]90[DisDllShutdownSR]==0, D18F4x1A8[MemTriStateEn], D18F3x84[Smaf4DramMemClkTri], D18F3x84[Smaf6DramMemClkTri] must be programmed to 1.

##### 2.5.6.1.1.1 System BIOS Requirements for Stutter Mode Operation During POST

BIOS creates a data structure in memory containing information about the processor for use by the driver. Please see your AMD representative for more information.

#### 2.5.6.2 EVENT_L

EVENT_L is a level sensitive input to the processor. When asserted, the actions specified by D18F2xA4 are taken. EVENT_L is generally asserted to indicate that a DRAM high temperature condition exists. The minimum assertion time for EVENT_L is 15 ns. The minimum deassertion time for EVENT_L is 15 ns.

- EVENT_L is pulled to VDDIO on the motherboard.
- EVENT_L is ignored while:
  - PWROK is de-asserted.
  - RESET_L is asserted.
- BIOS must ensure that throttling is disabled (D18F2xA4[ThrottleEn[1:0]]=00b) until DRAM training is complete.
- See 2.9.7 [DRAM On DIMM Thermal Management].

### 2.5.7 System Power Management

#### 2.5.7.1 S-states

S-states are ACPI defined sleep states. S0 is the operational state. All other S-states are low-power states in which the various voltage rails in the system may or may not be powered. See the ACPI specification for descriptions of each S-state.

##### 2.5.7.1.1 ACPI Suspend to RAM State (S3)

The processor supports the ACPI-defined S3 state. Software is responsible for restoring the state of the processor's registers when resuming from S3. All registers in the processor that BIOS initialized during the initial boot must be restored. The method used to restore the registers is system specific.

During S3 entry, system memory enters self-refresh mode. Software is responsible for bringing memory out of self-refresh mode when resuming from S3. To bring memory out of self-refresh mode, see 2.9.3 [DCT/DRAM Initialization and Resume].

Many of the systemboard power planes for the processor are powered down during S3. Refer to the AMD Family 12h Processor Electrical Data Sheet, #41609 for the following:

- Power plane electrical requirements during S3.
- Power plane sequencing requirements on S3 entry and exit.
- System signal states for both inputs (e.g. PWROK and RESET_L) and outputs (e.g. VID[*], PSI_L bit, THERMTRIP_L, etc.) during S3.
- System signal sequencing requirements on S3 entry and exit.
- System management message sequencing on S3 entry and exit.

## 2.6 Performance Monitoring

The processor includes support for two methods of monitoring processor performance: performance monitor counters and instruction based sampling (IBS).

### 2.6.1 Performance Monitor Counters

The performance monitor counters are used by software to count specific events that occur in the processor. [The Performance Event Select Register (PERF_CTL[3:0])] MSRC001_00[03:00] and [The Performance Event Counter Registers (PERF_CTR[3:0])] MSRC001_00[07:04] specify the events to be monitored and how

they are monitored. All of the events are specified in 3.24 [Performance Counter Events].

### 2.6.2 Instruction Based Sampling (IBS)

IBS is a code profiling mechanism that enables the processor to select a random instruction fetch or micro-op after a programmed time interval has expired and record specific performance information about the operation. An interrupt is generated when the operation is complete as specified by [The IBS Control Register] MSRC001_103A. An interrupt handler can then read the performance information that was logged for the operation.

The IBS mechanism is split into two parts: instruction fetch performance controlled through [The IBS Fetch Control Register (IbsFetchCtl)] MSRC001_1030; and instruction execution performance controlled through [The IBS Execution Control Register (IbsOpCtl)] MSRC001_1033. Instruction fetch sampling provides information about instruction TLB and instruction cache behavior for fetched instructions. Instruction execution sampling provides information about micro-op execution behavior. The data collected for instruction fetch performance is different from the data collected for instruction execution performance. Support for the IBS feature is indicated by the CPUID Fn8000_0001_ECX[IBS].

Instruction fetch performance is profiled by recording the following performance information (see MSRC001_1030, MSRC001_1031, and MSRC001_1032 for details of the events) for the tagged instruction fetch:

- If the instruction fetch completed or was aborted.
- The number of clock cycles spent on the instruction fetched.
- If the instruction fetch hit or missed the instruction cache.
- If the instruction fetch hit or missed the L1 and L2 TLBs.
- The linear and physical address associated with the fetch.

Instruction execution performance is profiled by tagging one micro-op. Instructions that decode to more than one micro-op return different performance data depending upon which micro-op associated with the instruction is tagged. The following performance information (see MSRC001_1033, MSRC001_1034, MSRC001_1035, MSRC001_1036, MSRC001_1037, MSRC001_1038 and MSRC001_1039 for details of the events) is returned for the tagged micro-op:

- Branch status for branch micro-ops.
- The number clocks from when the micro-op was tagged until the micro-op retires.
- The number clocks from when the micro-op completes execution until the micro-op retires.
- Source information for DRAM, MMIO and IO access.
- If the operation was a load or store that missed the data cache.
- If the operation was a load or store that hit or missed the L1 and L2 TLBs.
- The linear and physical address associated with a load or store operation.

## 2.7 Configuration Space

PCI-defined configuration space was originally defined to allow up to 256 bytes of register space for each function of each device; these first 256 bytes are called base configuration space (BCS). It was expanded to support up to 4096 bytes per function; bytes 256 through 4095 are called extended configuration space (ECS). The processor includes configuration space registers located in both BCS and ECS.

Configuration space is accessed by the processor through two methods:
- IO-space configuration: IO instructions to addresses CF8h and CFCh.
  - Enabled through [The IO-Space Configuration Address Register] IOCF8[ConfigEn], which allows access to BCS.
  - Access to ECS enabled through [The Northbridge Configuration Register (NB_CFG)] MSRC001_001F[EnableCf8ExtCfg].
  - Use of IO-space configuration can be programmed to generate GP faults through [The Hardware Configuration Register (HWCR)] MSRC001_0015[IoCfgGpFault].
  - SMI trapping for these accesses is specified by [The IO Trap Control Register (SMI_ON_IO_TRAP_CTL_STS)] MSRC001_0054 and [The IO Trap Registers (SMI_ON_IO_TRAP_[3:0])] MSRC001_00[53:50].
- MMIO configuration: configuration space is a region of memory space.
  - The base address and size of this range is specified by [The MMIO Configuration Base Address] MSRC001_0058. The size is controlled by the number of configuration-space bus numbers supported by the system. Accesses to this range are converted configuration space as follows:
    - Address[31:0] = {0000b, bus[7:0], device[4:0], function[2:0], offset[11:0]}.

### 2.7.1 MMIO Configuration Coding Requirements

MMIO configuration space is normally specified to be the uncacheable (UC) memory type. Instructions used to read MMIO configuration space are required to take the following form:

```
mov eax/ax/al, <any_address_mode>;
```

Instructions used to write MMIO configuration space are required to take the following form:

```
mov <any_address_mode>, eax/ax/al;
```

No other source/target registers may be use other than eax/ax/al.

In addition, all such accesses are required not to cross any naturally aligned DW boundary. Access to MMIO configuration space registers that do not meet these requirements result in undefined behavior.

### 2.7.2 MMIO Configuration Ordering

Since MMIO configuration cycles are not serializing in the way that IO configuration cycles are, their ordering rules relative to posted may result in unexpected behavior.

Therefore, processor MMIO configuration space is designed to match the following ordering relationship that exists naturally with IO-space configuration: if a CPU generates a configuration cycle followed by a posted-write cycle, then the posted write is held in the processor until the configuration cycle completes. As a result, any unexpected behavior that might have resulted if the posted-write cycle were to pass MMIO configuration cycle is avoided.

### 2.7.3 Processor Configuration Space

The processor includes configuration space as described in 3 [Registers]. Accesses to unimplemented registers of implemented functions are ignored: writes dropped; reads return 0's. Accesses to unimplemented functions also ignored: writes are dropped; however, reads return all F's. The processor does not log any master abort events for accesses to unimplemented registers or functions.

Accesses to device numbers of devices not implemented in the processor are routed based on the configuration map registers. If such requests are master aborted, then the processor can log the event.

## 2.8 The Northbridge (NB)

The processor includes a single NB that provides the interface to the core(s), system memory, and system IO devices. The NB includes all power planes except VDD; see 2.5.1 [Processor Power Planes And Voltage Control].

The NB is responsible for routing transactions sourced from cores and link to the appropriate core, cache, DRAM, or link. See 2.4.3 [Access Type Determination].

### 2.8.1 Northbridge (NB) Architecture

Major NB blocks are: North Bridge Front End (FRE), North Bridge Back End (XBR) and the DRAM Controller (DCT). The FRE interfaces with the core(s). The DCT maintains cache coherency and maintains a queue of incoming requests. The XBR is a switch that routes packets between the FRE, the DCT, and the link.

### 2.8.2 Northbridge Buffer Allocation Recommendations

**Table 8. Recommended buffer settings**

| Condition | D18F3x6C | | | | | | D18F3x74 | | | | | | D18F3x7C | | | | D18F3x17C | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | UpHiRespDBC | UpHiNpreqDBC | UpHiPreqDBC | UpLoRespDBC | UpLoNpreqDBC | UpLoPreqDBC | UpHiRespCBC | UpHiNpreqCBC | UpHiPreqCBC | UpLoRespCBC | UpLoNpreqCBC | UpLoPreqCBC | FreePoolBC | LoPriNpBC | LoPriPBC | CpuBC | HiPriNPBC | HiPriPBC |
| D0F0x98_x1E[HiPriEn]==0 | 0h | 0h | 0h | 1h | 1h | Eh | 0h | 0h | 0h | 8h | 9h | 7h | 19h | 1h | 1h | 1h | 0h | 0h |
| D0F0x98_x1E[HiPriEn]==1 | 0h | 1h | 0h | 1h | 1h | Dh | 0h | 1h | 0h | 8h | 8h | 7h | 18h | 1h | 1h | 1h | 1h | 0h |

### 2.8.3 DMA Exclusion Vectors (DEV)

The DEV is a set of protection tables in system memory that inhibit IO accesses to ranges of system memory. The tables specify link-defined UnitIDs or RequesterID's (Bus, Device, Function) that are allowed access to physical memory space on a 4 Kbyte page basis. Multiple protection domains are supported, each with independent DEV tables and supported UnitIDs/RequesterID's. See [The DEV Capability Header Register] D18F3xF0.

### 2.8.4 Northbridge Routing

#### 2.8.4.1 Address Space Routing

There are four main types of address space routed by the NB: (1) memory space targeting system DRAM, (2) memory space targeting IO (MMIO), (3) IO space, and (4) configuration space. The NB routing registers are accessed through function 1, offsets 40 through F4.

##### 2.8.4.1.1 DRAM and MMIO Memory Space

For memory-space transactions, the physical address, cacheability type, access type, and DRAM/MMIO desti-

nation type (see 2.4.4.1.2 [Determining The Access Destination for CPU Accesses]) are presented to the NB for further processing as follows:

- IO-device accesses are processed as follows:
  - If the access matches [The Memory Mapped IO Base Registers] D18F1x[B8,B0,A8,A0,98,90,88,80], then the transaction is routed to the root complex;
  - Else, if the access matches [The DRAM Base Register] D18F1x40, then the access is routed to the DCT;
  - Else, the access is routed to the UMI.
- For core accesses the routing is determined based on the DRAM/MMIO destination:
  - If the destination is DRAM:
    - If the access matches D18F1x40, then the transaction is routed to the DCT;
    - Else, the access is routed to the UMI.
  - If the destination is MMIO:
    - If the access matches D18F1x[B8,B0,A8,A0,98,90,88,80], then the transaction is routed to the root complex;
    - Else, the access is routed to the UMI.

#### 2.8.4.1.2 IO Space

IO-space transactions from the link or cores are routed as follows:

- If the access matches [The IO-Space Base Register] D18F1xC0, then the transaction is routed to root complex;
- Else, the access is routed to the UMI.

### 2.8.5 Physical Address Space

The processor supports 40 address bits of coherent memory space (1 terabyte) as indicated by [The Long Mode Address Size Identifiers] CPUID Fn8000_0008_EAX. The processor master aborts the following upper-address transactions (to address PhysAddr):

- Link requests with non-zero PhysAddr[63:40].

## 2.9 DRAM Controllers (DCTs)

The processor includes two DRAM controllers (DCTs). Each DCT controls one 64-bit DDR3 DRAM channel. A DRAM channel consists of the group of DRAM interface pins connecting to one series of DIMMs. DCT0 controls channel A and DCT1 controls channel B.

The DCTs operate on physical addresses translated into normalized addresses corresponding to the values programmed into [The DRAM CS Base Address Registers] D18F2x[1,0][4C:40]. Normalized addresses only include address bits within a DCT's range. The physical to normalized address translation varies based on DCT interleave and hoisting settings. See 2.9.4 [Memory Interleaving Modes] and 2.9.5 [Memory Hoisting].

The following restrictions limit the DIMM types and configurations supported by the DCTs:

- All DIMMs connected to a processor are required to operate at the same MEMCLK frequency, regardless of which channel they are connected to. Both DCTs must be programmed to the same frequency.
- Registered DIMMs are not supported.
- x4 (by 4) DIMMs are not supported.
- Quad rank DIMMs are not supported.

The tables below list the maximum DIMM speeds supported by the processor for different configurations. The motherboard should comply with the relevant AMD socket motherboard design guideline (MBDG) to achieve

the rated speeds. In cases where MBDG design options exist, lower-quality options may compromise the maximum achievable speed; motherboard designers should assess the tradeoffs.

**Table 9: DCT Definitions**

| Term | Definition |
|---|---|
| **DdrRate** | The DDR data rate (MT/s). |
| **DIMM0** | DIMM slots 0-n. The DIMMs on each channel are numbered from 0 to n where DIMM0 is the DIMM closest to the processor on that channel and DIMMn is the DIMM farthest from the processor on that channel. |
| **DIMM1** | |
| **DimmsPopulated** | The number of DIMMs populated per channel |
| **DR** | Dual Rank |
| **NP** | No DIMM populated |
| **NumDimmSlots** | The number of motherboard DIMM slots per channel |
| **SODIMM** | Small outline DIMM |
| **SR** | Single Rank |
| **UDIMM** | Unbuffered DIMM |
| **VDDIO** | DDR VDDIO in volts. |

**Table 10: DDR3 UDIMM Maximum Frequency Support for FM1**

| DIMM Slots/Ch | DIMMs | DIMMs | | Frequency[1] (MT/s) |
|---|---|---|---|---|
| | | SR | DR | 1.5V |
| 1 | 1 | 1 | - | 1866 |
| | | - | 1 | 1866 |
| 2 | 1 | 1 | - | 1600 |
| | | - | 1 | 1600 |
| | 2 | 2 | - | 1600 |
| | | 1 | 1 | 1333 |
| | | - | 2 | 1333 |

1. Population restrictions (including the order for partially populated channels) may apply. See Table 13.
2.

**Table 11: DDR3 SO-DIMM Maximum Frequency Support for FM1**

| DIMM Slots/Ch | DIMMs | DIMMs | | Frequency[1] (MT/s) |
|---|---|---|---|---|
| | | SR | DR | 1.5V |
| 1 | 1 | 1 | - | 1600 |
| | | - | 1 | 1600 |

**Table 11: DDR3 SO-DIMM Maximum Frequency Support for FM1**

| DIMM Slots/Ch | DIMMs | DIMMs | | Frequency[1] (MT/s) |
|---|---|---|---|---|
| | | SR | DR | 1.5V |
| 2 | 1 | 1 | - | 1333 |
| | | - | 1 | 1333 |
| | 2 | 2 | - | 1333 |
| | | 1 | 1 | 1333 |
| | | - | 2 | 1333 |

1. Population restrictions (including the order for partially populated channels) may apply. See Table 13.

**Table 12: DDR3 SO-DIMM Maximum Frequency Support for FS1**

| DIMM Slots/Ch | DIMMs | DIMMs | | Frequency[1] (MT/s) | |
|---|---|---|---|---|---|
| | | SR | DR | 1.5V | 1.35V |
| 1 | 1 | 1 | - | 1600 | 1333 |
| | | - | 1 | 1600 | 1333 |

1. Population restrictions (including the order for partially populated channels) may apply. See Table 14.

The tables below list the DIMM populations as supported by the processor. DIMMs must be populated from farthest slot to closest slot to the processor on a per channel basis when a daisy chain topology is used.

**Table 13: DDR3 UDIMM and SO-DIMM Population Support for FM1**

| DIMM Slots/Ch | DIMM0 | DIMM1 |
|---|---|---|
| 1 | SR/DR | N/A |
| 2 | - | SR/DR |
| | SR/DR | SR/DR |

**Table 14: DDR3 SO-DIMM Population Support for FS1**

| DIMM Slots/Ch | DIMM0 | DIMM1 |
|---|---|---|
| 1 | SR/DR | N/A |

### 2.9.1 DCT Configuration Registers

The DCT configuration registers reside in device 18h function 2 configuration space. For per DCT configuration registers, D18F2x0xx registers are associated with DCT0 and D18F2x1XX registers are associated with DCT1.

A subset of DCT configuration registers must be programmed for each supported NB P-state. See 2.9.3.4.7 [NB P-states for DCT/Dram Initialization and Training].

### 2.9.2 DRAM Controller Direct Response Mode

The DCT supports direct response mode for responding to a cache line fill request before the DCT is initialized. In direct response mode, the target DCT responds to a cache line fill request by returning 64 bytes of all ones without issuing a read transaction on the DRAM bus. The BIOS uses this feature to allocate cache lines for temporary data storage. The controller exits direct response mode when D18F2x[1,0]7C[EnDramInit] is set to 1.

See 2.9.3.6 [DRAM Device Initialization] and 2.3.3 [Cache Initialization For General Storage During Boot]

### 2.9.3 DCT/DRAM Initialization and Resume

DRAM initialization requires several steps to configure the DCTs and DIMMs, as well as tuning the DRAM channel to ensure stability. DRAM resume requires several steps to configure the DCTs to properly resume from the S3 state. The following sequence describes the steps needed to enable a DRAM channel after a reset for initialization or resume. To disable an unused DRAM channel see 2.9.3.8 [DRAM Channel Disable].

1. Configure the DDR supply voltage regulator. See 2.9.3.1.
2. Force NB P-state to NBP0. See 2.5.4.1.3.
   A. Program D18F6x90[NbPsCtrlDis]=1.
   B. Program D18F6x90[NbPsForceSel]=0.
   C. Program D18F6x90[NbPsForceReq]=1.
   D. Wait for D18F6x98[NbPs1Act]=0.
3. DDR phy initialization. See 2.9.3.2.
4. DRAM device and controller initialization.
   - If BIOS is booting from an unpowered state (ACPI S4, S5 or G3), then it performs the following:
     a. Program SPD configuration. See 2.9.3.3.
     b. Program Non-SPD configuration. See 2.9.3.4.
     c. Program NBP0 specific configuration. See 2.9.3.4.7.
     d. DRAM device initialization. See 2.9.3.6.
     e. Program DCT training specific configuration. See 2.9.3.5.
   - If BIOS is resuming the platform from S3 state, then it performs the following:
     a. Restore all DCT and phy registers that were programmed during the first boot from non-volatile storage, including NBP0 and NBP1 version of registers specified in 2.9.3.4.7. Use D18F6x98[NbPsDbgEn, NbPsCsrAccSel] to restore NBP0/NBP1 specific registers. See 2.9.3.3, 2.9.3.4, and 2.9.3.4.7 for a review of registers.
     b. Program D18F2x[1,0]90[ExitSelfRef] = 1.
     c. Restore the trained delayed values (found during the initial boot in steps 4 and 5 below) from non-volatile storage.
     d. Continue at step 12.
5. DRAM write levelization training. See 2.9.3.7.1.
6. DRAM data training.
   A. DQS receiver enable training. See 2.9.3.7.2.
   B. Program D18F2x[1,0]9C_x0D0F_E003[DisAutoComp, DisablePredriverCal] = {0b, 1b}.
   C. Initialize Receive FIFO pointers:
      a. Program D18F2x[1,0]78[RxPtrInitReq]=1.
      b. Wait for D18F2x[1,0]78[RxPtrInitReq]=0.
   D. Program D18F2x[1,0]A8[DbeGskMemClkAlignMode] as follows:
      a. Program D18F2x[1,0]90[DisDllShutDownSR] = 1.
      b. Program D18F2x[1,0]90[EnterSelfRef] = 1.
      c. Wait for D18F2x[1,0]90[EnterSelfRef] = 0.

d. Program D18F2x[1,0]A8[DbeGskMemClkAlignMode] = 10b.
e. Program D18F2x[1,0]90[ExitSelfRef] = 1.
f. Wait for D18F2x[1,0]90[ExitSelfRef] = 0.
g. Program D18F2x[1,0]90[DisDllShutDownSR] = 0.

E. DQS receiver enable cycle training. See 2.9.3.7.3.
F. DQS position training. See 2.9.3.7.4.
G. MaxRdLatency training. See 2.9.3.7.5.1.

IF (D18F6x90[NbPsCap]=1) THEN
7. Program NBP1 specific configuration. See 2.9.3.4.7. Use D18F6x98[NbPsDbgEn, NbPsCsrAccSel] to select NBP1 registers.
8. IF (D18F2x[1,0]94[MemClkFreq] <= 667 MHz) THEN
Program D18F6x90[NbPs1Vid] = D18F6x90[NbPs1Vid] + FCRxFE00_705F[GnbIdleAdjustVid].
ENDIF.
9. Force NB P-state to NBP1. See 2.5.4.1.3.
A. Program D18F6x90[NbPsForceSel]=1.
B. Wait for D18F6x98[NbPs1Act]=1.
10. MaxRdLatency training for NBP1. See 2.9.3.7.5.1.
11. IF (D18F2x[1,0]94[MemClkFreq] <= 667 MHz) THEN
Program D18F6x90[NbPs1Vid] = D18F6x90[NbPs1Vid] - FCRxFE00_705F[GnbIdleAdjustVid].
ENDIF.

ENDIF.
12. Release NB P-state force. See 2.5.4.1.3.
A. Program D18F6x90[NbPsForceReq]=0.
B. Program D18F6x90[NbPsCtrlDis]=0.
13. Program DCT for normal operation. See 2.9.3.5.
14. Program DRAM Phy for power savings. See 2.9.3.9.

The DRAM subsystem is ready for use.

#### 2.9.3.1 Low Voltage DDR3

JEDEC defined 1.5V and 1.35V DDR3 devices are supported. Platforms that support 1.35V operation should power on VDDIO at 1.35V until operating voltage is determined by reading the SPD ROM of all the DIMMs. BIOS should not operate DIMMs at voltages higher than supported as indicated by the SPD.

The recommended BIOS configuration sequence is as follows:

1. BIOS reads the SPD ROM of all DIMMs to determine the common operating voltages.
2. BIOS configures VDDIO to match the lowest common supported voltage based on the SPD values. See platform specific documentation for changing the voltage.

#### 2.9.3.2 DDR Phy Initialization

The BIOS initializes the phy and the internal interface from the DCT to the phy, including the PLLs and the fence value, after each reset and for each time a frequency change is made.

BIOS obtains size, loading, and frequency information about the DIMMs and channels using SPDs prior to phy initialization. BIOS then performs the following actions:

1. Program D18F2x[1,0]9C_x0000_000B = 80000000h.
2. Program D18F2x[1,0]9C_x0D0F_E013[PllRegWaitTime] = 0118h.

3. Phy Voltage Level Programming. See 2.9.3.2.1.
4. DRAM channel frequency change. See 2.9.3.2.2.
5. If BIOS is booting from an unpowered state (ACPI S4, S5 or G3; not S3, suspend to RAM), then it performs the following:
   A. Program D18F2x[1,0]A8[DbeGskMemClkAlignMode] = 00b.
   B. Program D18F2x[1,0]7C[EnDramInit] = 1.
6. Phy fence programming. See 2.9.3.2.3.
7. Phy compensation initialization. See 2.9.3.2.4.

#### 2.9.3.2.1 Phy Voltage Level Programming

BIOS programs the following according to the desired phy VDDIO voltage level:

- Program D18F2x[1,0]9C_x0D0F_0[F,7:0]1F[RxVioLvl].
- Program D18F2x[1,0]9C_x0D0F_[C,8,2][F,1:0]1F[RxVioLvl].
- Program D18F2x[1,0]9C_x0D0F_4009[CmpVioLvl].

See 2.9.3.1 [Low Voltage DDR3].

#### 2.9.3.2.2 DRAM Channel Frequency Change

The following sequence is used to change the DRAM frequency under all boot conditions, including restoring the DCT state when resuming from the S3 state:

For each DCT:
1. Program D18F2x[1,0]9C_x0D0F_E006[PllLockTime] = 0190h.
2. Program D18F2x[1,0]94[MemClkFreqVal] = 0.
3. Program D18F2x[1,0]94[MemClkFreq] to the desired DRAM frequency.
4. Setup NBP0 and NBP1 frequency and voltage to meet NCLK-MEMCLK ratio requirements for the desired DRAM frequency. See 2.5.4.1.1 [BIOS Requirements for NB P-state Initialization During DRAM Training].

For each DCT:
5. Program the following according to the new MemClkFreq value:
   A. Program D18F2x[1,0]F4_x30[DbeGskFifoNumerator] = NclkFid * MemClkDid * 16.
      - NclkFid = NCLK PLL multiplier as defined by D18F3xD4[MainPllOpFreqId](i.e. COF/100MHz).
      - MemClkDid = MEMCLK PLL divide ratio as defined by Table 79.
      -
   B. Program D18F2x[1,0]F4_x31[DbeGskFifoDenominator] = NclkDiv * PllMult;
      - NclkDiv = If programming NBP1 then D18F6x90[NbPs1NclkDiv], else D18F3xDC[NbPs0NclkDiv].
      - PllMult = MEMCLK PLL multiplier as defined by Table 79.
   C. Program D18F2x[1,0]F4_x32[DataTxFifoSchedDlyNegSlot1, DataTxFifoSchedDlySlot1, DataTxFifoSchedDlyNegSlot0, DataTxFifoSchedDlySlot0]. See 2.9.3.2.2.2 [DCT Transmit Fifo Schedule Delay Programming].
   D. Program D18F2x[1,0]78[RdPtrInit] = 7.
   E. Program D18F2x[1,0]9C_x0D0F_0[F,7:0]13[ProcOdtAdv] = IF (SODIMM && (D18F2x[1,0]94[MemClkFreq] <= 667 MHz)) THEN 0 ELSE 1 ENDIF.

For each DCT:
6. Program D18F2x[1,0]94[MemClkFreqVal] = 1.

For each DCT:

7. IF (D18F2x[1,0]9C_x0D0F_E00A[CsrPhySrPllPdMode]==0) THEN program D18F2x[1,0]9C_x0D0F_E006[PllLockTime] = 0Fh.

#### 2.9.3.2.2.1 Requirements for DRAM Frequency Change During Training

During DRAM training, BIOS may be required to change the DRAM(MEMCLK) frequency. The steps below describe what is required to prepare the processor and memory subsystem for the new MEMCLK frequency. It is assumed that the memory subsystem has previously been initialized at the current MEMCLK frequency, and this procedure describes only the steps that must be repeated at the new MEMCLK frequency. See 2.9.3.7.1 [Write Levelization Training] and 2.9.3.7.2 [DQS Receiver Enable Training].

1. Ensure NB P-states are disabled prior to this procedure. See D18F6x90[NbPsCtrlDis].
2. Enter self-refresh:
   A. Program D18F2x[1,0]90[DisDllShutDownSR] = 1.
   B. Program D18F2x[1,0]90[EnterSelfRef] = 1.
   C. Wait for D18F2x[1,0]90[EnterSelfRef] = 0.
3. DRAM channel frequency change. See 2.9.3.2.2.
4. Exit self-refresh:
   A. Program D18F2x[1,0]90[ExitSelfRef] = 1.
   B. Wait for D18F2x[1,0]90[ExitSelfRef] = 0.
   C. Program D18F2x[1,0]90[DisDllShutDownSR] = 0.
5. Phy fence programming. See 2.9.3.2.3.
6. Phy compensation initialization. See 2.9.3.2.4.
7. Program SPD configuration. See 2.9.3.3.
8. Program Non-SPD configuration. See 2.9.3.4.
9. Program NB P-state specific configuration. See 2.9.3.4.7.
10. Issue MRS(2), MRS(3), MRS(1), MRS(0) commands. See 2.9.3.6.1 [Software DDR3 Device Initialization].

#### 2.9.3.2.2.2 DCT Transmit Fifo Schedule Delay Programming

The optimal value for D18F2x[1,0]F4_x32[DataTxFifoSchedDlyNegSlot1, DataTxFifoSchedDlySlot1, DataTxFifoSchedDlyNegSlot0, DataTxFifoSchedDlySlot0] is configuration specific. BIOS should use the guidelines below to configure the recommended values:

For N=0,1:

- If (PartialSumSlotN >= 0):
  - DataTxFifoSchedDlySlotN=CEIL(PartialSumSlotN).
  - DataTxFifoSchedDlyNegSlotN=0.
- Else if (PartialSumSlotN < 0):
  - DataTxFifoSchedDlySlotN=ABS(CEIL(PartialSumSlotN*MemClkPeriod/NclkPeriod)).
  - DataTxFifoSchedDlyNegSlotN=1.

- PartialSumSlot0 = ((7 * NclkPeriod[1]) + (1.5 * MemClkPeriod[2]) + 520ps)*MemClkFrequency - tCWL[3] - CmdSetup[4] - SlowAccessMode[5] - PtrSeparation[6].
- PartialSumSlot1 = ((7 * NclkPeriod[1]) + (1.5 * MemClkPeriod[2]) + 520ps)*MemClkFrequency - tCWL[3] - CmdSetup[4] - 1 - PtrSeparation[6] - SlowAccessMode[5].

1. NclkPeriod = NCLK period as defined by D18F3xDC[NbPs0NclkDiv] or D18F6x90[NbPs1NclkDiv] for

given NB P-state.
2. MemClkPeriod/MemClkFrequency = MEMCLK period/frequency as defined by D18F2x[1,0]94[MemClkFreq].
3. tCWL = Tcwl in MEMCLKs as defined by D18F2x[1,0]84[Tcwl].
4. CmdSetup = 1/2 MEMCLK if all of D18F2x[1,0]9C_x0000_0004[AddrCmdSetup, CsOdtSetup, CkeSetup]=0, else 1 MEMCLK.
5. SlowAccessMode = 1 MEMCLK if D18F2x[1,0]94[SlowAccessMode]=1, else 0.
6. PtrSeparation:
   - PtrSeparation = ((16 + RdPtrInitMin - D18F2x[1,0]78[RdPtrInit]) MOD 16)/2 + RdPtrInitRmdr.
   - RdPtrInitMin = 7.
   - If (D18F2x[1,0]94[MemClkFreq] >= 800 MHz)
     then RdPtrInitRmdr = (((4.5 * MemClkPeriod) - 990ps) MOD MemClkPeriod)/MemClkPeriod
     else RdPtrInitRmdr = (((4.5 * MemClkPeriod) - 1466ps) MOD MemClkPeriod)/MemClkPeriod

#### 2.9.3.2.3 Phy Fence Programming

The DDR phy fence logic is used to adjust the phase relationship between the data FIFO and the data going to the pad. After any MEMCLK frequency change and before any memory training, BIOS must perform phy fence training using the following steps:

For each channel:
1. Program D18F2x[1,0]9C_x0000_0008[FenceTrSel]=10b.
2. Program D18F2x[1,0]9C_x0000_00[51:50]=1313_1313h.
3. Perform phy fence training. See 2.9.3.2.3.1 [Phy Fence Training].
4. Write the calculated fence value to D18F2x[1,0]9C_x0000_000C[FenceThresholdTxDll].
5. Program D18F2x[1,0]9C_x0D0F_0[F,7:0]0F[AlwaysEnDllClks]=001b.
6. Program D18F2x[1,0]9C_x0000_0008[FenceTrSel]=01b.
7. Program D18F2x[1,0]9C_x0000_00[51:50]=1313_1313h.
8. Perform phy fence training. See 2.9.3.2.3.1 [Phy Fence Training].
9. Write the calculated fence value to D18F2x[1,0]9C_x0000_000C[FenceThresholdRxDll].
10. Program D18F2x[1,0]9C_x0D0F_0[F,7:0]0F[AlwaysEnDllClks]=000b.
11. Program D18F2x[1,0]9C_x0000_0008[FenceTrSel]=11b.
12. Program D18F2x[1,0]9C_x0000_00[51:50]=1313_1313h.
13. Perform phy fence training. See 2.9.3.2.3.1 [Phy Fence Training].
14. Write the calculated fence value to D18F2x[1,0]9C_x0000_000C[FenceThresholdTxPad].
15. IF (D18F2x[1,0]9C_x0000_000C[FenceThresholdTxPad] < 16) THEN
        Program D18F2x[1,0]9C_x0D0F_[C,8,2][F,1:0]31 = {001h, D18F2x[1,0]9C_x0000_000C[19:16]}
    ELSE
        Program D18F2x[1,0]9C_x0D0F_[C,8,2][F,1:0]31 = 0000h
    ENDIF.
16. Program Fence2 threshold for data as follows:
    A. IF (D18F2x[1,0]9C_x0000_000C[FenceThresholdTxPad] < 16) THEN
           Fence2_TxPad[4:0] = {1b, D18F2x[1,0]9C_x0000_000C[19:16]}
       ELSE
           Fence2_TxPad[4:0] = 00000b
       ENDIF.
    B. IF (D18F2x[1,0]9C_x0000_000C[FenceThresholdRxDll] < 16) THEN
           Fence2_RxDll[4:0] = {1b, D18F2x[1,0]9C_x0000_000C[24:21]}
       ELSE
           Fence2_RxDll[4:0] = 00000b
       ENDIF.

C. IF (D18F2x[1,0]9C_x0000_000C[FenceThresholdTxDll] < 16) THEN
   Fence2_TxDll[4:0] = {1b, D18F2x[1,0]9C_x0000_000C[29:26]}
   ELSE
   Fence2_TxDll[4:0] = 00000b
   ENDIF.
D. Program D18F2x[1,0]9C_x0D0F_0[F,7:0]31 = {0b, Fence2_RxDll[4:0], Fence2_TxDll[4:0], Fence2_TxPad[4:0]}.

17. Reprogram D18F2x[1,0]9C_x0000_0004.

When resuming from S3, it is recommended that BIOS reprogram D18F2x[1,0]9C_x0000_000C[FenceThresholdTxDll, FenceThresholdRxDll, FenceThresholdTxPad], D18F2x[1,0]9C_x0D0F_[C,8,2][F,1:0]31, and D18F2x[1,0]9C_x0D0F_0[F,7:0]31 from values stored in non-volatile storage instead of training.

#### 2.9.3.2.3.1 Phy Fence Training

The following describes the steps for each pass of phy fence training:

1. Program D18F2x[1,0]9C_x0000_0008[PhyFenceTrEn]=1.
2. Wait 2000 MEMCLKs.
3. Program D18F2x[1,0]9C_x0000_0008[PhyFenceTrEn]=0.
4. Read the phase recovery engine registers D18F2x[1,0]9C_x[51:50].
5. Calculate the fence value by averaging the fine delay values of all byte lanes and subtract 6.

#### 2.9.3.2.4 Phy Compensation Initialization

Each DDR IO driver has a programmable slew rate controlled by the pre-driver calibration code. The recommended slew rate is a function of the DC drive strength. BIOS initializes the recommended nominal slew rate values as follows:

1. Program D18F2x[1,0]9C_x0D0F_E003[DisAutoComp, DisablePreDriverCal] = {1b, 1b}.
2. Program TxPreP/TxPreN for Data and DQS according to Table 15 if VDDIO is 1.5V or Table 16 if 1.35V.
   A. Program D18F2x[1,0]9C_x0D0F_0[F,7:0]0[A,6]={0000b, TxPreP, TxPreN}.
   B. Program D18F2x[1,0]9C_x0D0F_0[F,7:0]02={1000b, TxPreP, TxPreN}.
3. Program TxPreP/TxPreN for Cmd/Addr according to Table 17 if VDDIO is 1.5V or Table 18 if 1.35V.
   A. Program D18F2x[1,0]9C_x0D0F_[C,8][1:0][12,0E,0A,06]={0000b, TxPreP, TxPreN}.
   B. Program D18F2x[1,0]9C_x0D0F_[C,8][1:0]02={1000b, TxPreP, TxPreN}.
4. Program TxPreP/TxPreN for Clock according to Table 19 if VDDIO is 1.5V or Table 20 if 1.35V.
   A. Program D18F2x[1,0]9C_x0D0F_2[1:0]02={1000b, TxPreP, TxPreN}.

**Table 15: Phy predriver calibration codes for Data/DQS at 1.5V**

| DDR Rate | Drive Strength[1] | TxPreP[2] | TxPreN[2] |
|---|---|---|---|
| 800 | 000b | 10_0100b | 10_0100b |
| | 001b | 10_0100b | 10_0100b |
| | 010b | 10_0100b | 10_0100b |
| | 011b | 10_0100b | 10_0100b |
| 1066 - 1333 | 000b | 11_1111b | 11_0110b |
| | 001b | 11_1111b | 11_0110b |
| | 010b | 11_1111b | 11_0110b |
| | 011b | 11_1111b | 11_0110b |

**Table 15: Phy predriver calibration codes for Data/DQS at 1.5V**

| DDR Rate | Drive Strength[1] | TxPreP[2] | TxPreN[2] |
|---|---|---|---|
| 1600 - 1866 | 000b | 11_1111b | 11_0110b |
| | 001b | 11_1111b | 11_0110b |
| | 010b | 11_1111b | 11_0110b |
| | 011b | 11_1111b | 11_0110b |

1. IF (D18F2x[1,0]9C_x0D0F_0[F,7:0]06) THEN
   See D18F2x[1,0]9C_x0000_0000[DqsDrvStren]
   ELSE
   See D18F2x[1,0]9C_x0000_0000[DataDrvStren]
   ENDIF.
2. See D18F2x[1,0]9C_x0D0F_0[F,7:0]0[A,6] and D18F2x[1,0]9C_x0D0F_0[F,7:0]02.

**Table 16: Phy predriver calibration codes for Data/DQS at 1.35V**

| DDR Rate | Drive Strength[1] | TxPreP[2] | TxPreN[2] |
|---|---|---|---|
| 800 | 000b | 11_1111b | 11_0110b |
| | 001b | 10_1101b | 10_1101b |
| | 010b | 10_1101b | 10_1101b |
| | f011b | 10_0100b | 10_0100b |
| 1066 - 1333 | 000b | 11_1111b | 11_0110b |
| | 001b | 11_1111b | 11_0110b |
| | 010b | 11_1111b | 11_0110b |
| | 011b | 11_1111b | 11_0110b |

1. IF (D18F2x[1,0]9C_x0D0F_0[F,7:0]06) THEN
   See D18F2x[1,0]9C_x0000_0000[DqsDrvStren]
   ELSE
   See D18F2x[1,0]9C_x0000_0000[DataDrvStren]
   ENDIF.
2. See D18F2x[1,0]9C_x0D0F_0[F,7:0]0[A,6] and D18F2x[1,0]9C_x0D0F_0[F,7:0]02.

**Table 17: Phy predriver calibration codes for Cmd/Addr at 1.5V**

| DDR Rate | Drive Strength[1] | TxPreP[2] | TxPreN[2] |
|---|---|---|---|
| 800 | 000b | 01_0010b | 01_0010b |
| | 001b | 01_0010b | 01_0010b |
| | 010b | 01_0010b | 01_0010b |
| | 011b | 01_0010b | 01_0010b |
| 1066 - 1333 | 000b | 01_1011b | 01_1011b |
| | 001b | 01_1011b | 01_1011b |
| | 010b | 01_1011b | 01_1011b |
| | 011b | 01_1011b | 01_1011b |

**Table 17: Phy predriver calibration codes for Cmd/Addr at 1.5V**

| DDR Rate | Drive Strength[1] | TxPreP[2] | TxPreN[2] |
|---|---|---|---|
| 1600 - 1866 | 000b | 10_1101b | 10_1101b |
| | 001b | 10_1101b | 10_1101b |
| | 010b | 10_1101b | 10_1101b |
| | 011b | 10_1101b | 10_1101b |

1. IF (D18F2x[1,0]9C_x0D0F_C002)THEN
   See D18F2x[1,0]9C_x0000_0000[CkeDrvStren]
   ELSEIF (D18F2x[1,0]9C_x0D0F_800[A,6,2])THEN
   See D18F2x[1,0]9C_x0000_0000[CsOdtDrvStren]
   ELSE
   See D18F2x[1,0]9C_x0000_0000[AddrCmdDrvStren]
   ENDIF.
2. See D18F2x[1,0]9C_x0D0F_[C,8][1:0][12,0E,0A,06] and D18F2x[1,0]9C_x0D0F_[C,8][1:0]02.

**Table 18: Phy predriver calibration codes for Cmd/Addr at 1.35V**

| DDR Rate | Drive Strength[1] | TxPreP[2] | TxPreN[2] |
|---|---|---|---|
| 800 | 000b | 01_0010b | 01_0010b |
| | 001b | 01_0010b | 01_0010b |
| | 010b | 01_0010b | 01_0010b |
| | 011b | 01_0010b | 01_0010b |
| 1066 - 1333 | 000b | 10_0100b | 10_0100b |
| | 001b | 01_1011b | 01_1011b |
| | 010b | 01_1011b | 01_1011b |
| | 011b | 01_1011b | 01_1011b |

1. IF (D18F2x[1,0]9C_x0D0F_C002)THEN
   See D18F2x[1,0]9C_x0000_0000[CkeDrvStren]
   ELSEIF (D18F2x[1,0]9C_x0D0F_800[A,6,2])THEN
   See D18F2x[1,0]9C_x0000_0000[CsOdtDrvStren]
   ELSE
   See D18F2x[1,0]9C_x0000_0000[AddrCmdDrvStren]
   ENDIF.
2. See D18F2x[1,0]9C_x0D0F_[C,8][1:0][12,0E,0A,06] and D18F2x[1,0]9C_x0D0F_[C,8][1:0]02.

**Table 19: Phy predriver calibration codes for Clock at 1.5V**

| DDR Rate | Drive Strength[1] | TxPreP[2] | TxPreN[2] |
|---|---|---|---|
| 800 | 000b | 10_0100b | 10_0100b |
| | 001b | 10_0100b | 10_0100b |
| | 010b | 10_0100b | 10_0100b |
| | 011b | 10_0100b | 10_0100b |

**Table 19: Phy predriver calibration codes for Clock at 1.5V**

| DDR Rate | Drive Strength[1] | TxPreP[2] | TxPreN[2] |
|---|---|---|---|
| 1066 - 1333 | 000b | 11_1111b | 11_0110b |
| | 001b | 11_1111b | 11_0110b |
| | 010b | 11_1111b | 11_0110b |
| | 011b | 10_1101b | 10_1101b |
| 1600 - 1866 | 000b | 11_1111b | 11_0110b |
| | 001b | 11_1111b | 11_0110b |
| | 010b | 11_1111b | 11_0110b |
| | 011b | 11_1111b | 11_0110b |

1. See D18F2x[1,0]9C_x0000_0000[ClkDrvStren].
2. See D18F2x[1,0]9C_x0D0F_2[1:0]02.

**Table 20: Phy predriver calibration codes for Clock at 1.35V**

| DDR Rate | Drive Strength[1] | TxPreP[2] | TxPreN[2] |
|---|---|---|---|
| 800 | 000b | 11_0110b | 10_1101b |
| | 001b | 11_0110b | 10_1101b |
| | 010b | 10_0100b | 10_0100b |
| | 011b | 10_0100b | 10_0100b |
| 1066 - 1333 | 000b | 11_1111b | 11_0110b |
| | 001b | 11_1111b | 11_0110b |
| | 010b | 11_1111b | 11_0110b |
| | 011b | 11_0110b | 10_1101b |

1. See D18F2x[1,0]9C_x0000_0000[ClkDrvStren].
2. See D18F2x[1,0]9C_x0D0F_2[1:0]02.

#### 2.9.3.3 SPD ROM-Based Configuration

The Serial Presence Detect (SPD) ROM is a non-volatile memory device on the DIMM encoded by the DIMM manufacturer. The description of the SPD is usually provided on a data sheet for the DIMM itself along with data describing the memory devices used. The data describes configuration and speed characteristics of the DIMM and the SDRAM components mounted on the DIMM. The associated data sheet also contains the DIMM byte values that are encoded in the SPD on the DIMM.

BIOS reads the values encoded in the SPD ROM through a system-specific interface. BIOS acquires DIMM configuration information, such as the amount of memory on each DIMM, from the SPD ROM on each DIMM and uses this information to program the DRAM controller registers.

The SPD ROM provides values for several DRAM timing parameters that are required by the DCT. In general, BIOS should use the optimal value specified by the SPD ROM. These parameters are:

- D18F2x[1,0]84[Twr]: Write recovery time
- D18F2x[1,0]88[Tcl]: CAS latency
- D18F2x[1,0]8C[Tref]: Refresh interval

- D18F2x[1,0]8C[Trfc1, Trfc0]: Auto Refresh to Active/Auto Refresh delay
- D18F2x[1,0]94[FourActWindow]: Four activate window delay time
- D18F2x[1,0]F4_x40[Tras]: Active to Precharge delay
- D18F2x[1,0]F4_x40[Trc]: Active to Active/Auto Refresh delay
- D18F2x[1,0]F4_x40[Trcd]: RAS to CAS delay
- D18F2x[1,0]F4_x40[Trp]: Precharge time
- D18F2x[1,0]F4_x41[Trrd]: Active-Bank-A to Active-Bank-B delay
- D18F2x[1,0]F4_x41[Trtp]: Internal Read to Precharge command delay
- D18F2x[1,0]F4_x41[Twtr]: Internal write to read command delay

Optimal cycle time is specified for each DIMM and is used to limit or determine bus frequency. See 2.9.3.6 [DRAM Device Initialization].

#### 2.9.3.3.1 FourActWindow (Four Bank Activate Window or tFAW)

No more than 4 banks may be activated in a rolling tFAW window, as configured by D18F2x[1,0]94[FourActWindow]. To program this field, BIOS must convert the tFAW parameter into MEMCLK cycles by dividing the highest tFAW parameter (in ns) found in all the DIMMs connected to the channel by the period of MEMCLK (in ns) and rounding up to the next integer.

### 2.9.3.4 Non-SPD ROM-Based Configuration

There are several DRAM timing parameters and DCT configurations that need to be programmed for optimal memory performance. These values are not derived from the SPD ROM. Several of these timing parameters are functions of other configuration values. These interdependencies must be considered when programming values into several DCT register timing fields. The factors to consider when specifying a value for a specific non-SPD timing parameter are:

- Training delay values. See 2.9.3.7 [DRAM Training].
- Read and write latency differences.
- The phy's idle clock requirements on the data bus.
- DDR3 ODT timing requirements.
- NCLK frequency for each supported NB P-state.
- MEMCLK frequency.

The following subsections describe how BIOS programs each non-SPD related timing field to a recommended minimum timing value with respect to the above factors.

The following terms are defined to simplify calculations and are calculated in MEMCLKs:
- Latency Difference (LD) = D18F2x[1,0]88[Tcl] - D18F2x[1,0]84[Tcwl].
- Read ODT Delay (ROD) = MAX(0, D18F2x[1,0]F4_x83[RdOdtOnDuration] - 6).
- Write ODT Delay (WOD) = MAX(0, D18F2x[1,0]F4_x83[WrOdtOnDuration] - 6).

#### 2.9.3.4.1 Trdrd and TrdrdSD (Read-to-Read Timing)

The optimal values for D18F2x[1,0]8C[Trdrd] and D18F2x[1,0]F4_x06[TrdrdSD] are platform and configuration specific and should be characterized for best performance. Prior to DRAM training, BIOS should program these parameters to the largest defined value. After DRAM training, BIOS should use the guidelines below to configure the recommended platform generic timing values:

- TrdrdSD (in MEMCLKs) = 3.

- Trdrd (in MEMCLKs) = CEIL(MAX(ROD + 3, $CDD_{Trdrd}$ / 2 + 3.5)).

The Critical Delay Difference (CDD) is the largest delay difference of the channel.
- Each delay difference is D18F2x[1,0]9C_x0000_00[24:10][DqsRcvEnGrossDelay] minus D18F2x[1,0]9C_x0000_00[24:10][DqsRcvEnGrossDelay].
- For $CDD_{Trdrd}$, the subtraction terms are the delays of different DIMMs within the same byte lane.

BIOS must program these parameters as follows: TrdrdSD <= Trdrd.

#### 2.9.3.4.2 Twrwr and TwrwrSD (Write-to-Write Timing)

The optimal values for D18F2x[1,0]8C[Twrwr] and D18F2x[1,0]F4_x16[TwrwrSD] are platform and configuration specific and should be characterized for best performance. Prior to DRAM training, BIOS should program these parameters to the largest defined value. After DRAM training, BIOS should use the guidelines below to configure the recommended platform generic timing values:

- TwrwrSD (in MEMCLKs) = WOD + 3.
- Twrwr (in MEMCLKs) = CEIL(MAX(WOD + 3, $CDD_{Twrwr}$ / 2 + 3.5)).

The Critical Delay Difference (CDD) is the largest delay difference of the channel.
- Each delay difference is D18F2x[1,0]9C_x0000_00[44:30][WrDqsGrossDly] minus D18F2x[1,0]9C_x0000_00[44:30][WrDqsGrossDly].
- For $CDD_{Twrwr}$, the subtraction terms are the delays of different DIMMs within the same byte lane.

BIOS must program these parameters as follows: TwrwrSD <= Twrwr.

#### 2.9.3.4.3 Twrrd and TwrrdSD (Write-to-Read DIMM Termination Turn-around)

The optimal value for D18F2x[1,0]8C[Twrrd] and D18F2x[1,0]F4_x06[TwrrdSD] is platform and configuration specific and should be characterized for best performance. Prior to DRAM training, BIOS should program these parameters to the largest defined value. After DRAM training, BIOS should use the guidelines below to configure the recommended platform generic timing values:

- TwrrdSD (in MEMCLKs) = CEIL(MAX(1, MAX(WOD, $CDD_{TwrrdSD}$ / 2 + 0.5) - LD + 3)).
- Twrrd (in MEMCLKs) = CEIL(MAX(1, MAX(WOD, $CDD_{Twrrd}$ / 2 + 0.5) - LD + 3)).

The Critical Delay Difference (CDD) is the largest delay difference of the channel.
- Each delay difference is D18F2x[1,0]9C_x0000_00[44:30][WrDqsGrossDly] minus D18F2x[1,0]9C_x0000_00[24:10][DqsRcvEnGrossDelay].
- For $CDD_{TwrrdSD}$, the subtraction terms are the delays of the same DIMM within the same byte lane.
- For $CDD_{Twrrd}$, the subtraction terms are the delays of different DIMMs within the same byte lane.

BIOS must program these parameters as follows: TwrrdSD <= Twrrd.

#### 2.9.3.4.4 TrwtTO (Read-to-Write Turnaround for Data, DQS Contention)

The optimal value for D18F2x[1,0]8C[TrwtTO] is platform and configuration specific and should be characterized for best performance. Prior to DRAM training, BIOS should program this parameter to the largest defined value. After DRAM training, BIOS should use the guidelines below to configure the recommended platform generic timing values after DDR training is complete:

- TrwtTO (in MEMCLKs) = CEIL(MAX(ROD, $CDD_{TrwtTO}$ / 2 - 0.5) + LD + 3).
  - If 1 DIMM/ch, substitute ROD=0.

The Critical Delay Difference (CDD) is the largest delay difference of the channel.
- Each delay difference is D18F2x[1,0]9C_x0000_00[24:10][DqsRcvEnGrossDelay] minus D18F2x[1,0]9C_x0000_00[44:30][WrDqsGrossDly].
- For $CDD_{TrwtTO}$, the subtraction terms are the delays of all DIMMs within the same byte lane.

#### 2.9.3.4.5 DRAM ODT Control

This section describes the ODT configurations and settings for the processor and attached DIMMs. The tables specify ODT values for different configurations, on a per channel basis. The DIMM termination values are programmed as specified below during DDR3 device initialization. If the DIMM termination values are changed after device initialization then BIOS must issue MRS commands to the devices to change the values. See 2.9.3.6.1 [Software DDR3 Device Initialization].

Table 21 specifies the ODT nominal (non-write) and dynamic termination resistance values for different DIMM configurations.

BIOS configures the ODT turn on delay and duration for reads and writes. See D18F2x[1,0]F4_x83[RdOdtTrnOnDly, RdOdtOnDuration, WrOdtTrnOnDly, WrOdtOnDuration].

**Table 21: DIMM ODT settings**

| DDR Rate | DIMMs Populated[1] | DIMM ODT (Rtt_Nom) | DIMM Dynamic ODT (Rtt_Wr) |
|---|---|---|---|
| 800 | 1 | 120 ohms | Disabled |
| 1066 | 1 | 120 ohms | Disabled |
| 1333 | 1 | 60 ohms | Disabled |
| 1600 - 1866 | 1 | 40 ohms | Disabled |
| 800 | 2 | 40 ohms | 120 ohms |
| 1066 | 2 | IF SO-DIMM 30 ohms, ELSE 40 ohms | 120 ohms |
| 1333 | 2 | 30 ohms | 120 ohms |
| 1600 | 2 | 20 ohms | 60 ohms |
| 1. DIMMs can be single or dual rank. DIMMs can be unbuffered or SO- DIMMs. | | | |

The following describes the general ODT behavior for various system configurations. In all cases, the processor ODT is off for writes and is on for reads:

- For one single or dual rank DIMM on a channel:
  - For writes, the ODT is on for the target rank.
  - For reads, the ODT is off for all ranks.
- For two single or dual rank DIMMs on a channel:
  - For writes, the ODT is on for the target rank of the target DIMM and also on for the first rank of the non-target DIMM.

- For reads, the ODT is on for the first rank of the non-target DIMM.

BIOS configures the DIMM ODT behavior on a per chip select basis according to the DIMM population. The ODT patterns for reads and writes are programmed using D18F2x[1,0]F4_x180 and D18F2x[1,0]F4_x182, respectively, as specified by Table 22. BIOS also configures the DIMM ODT pattern used during write levelization training by setting D18F2x[1,0]9C_x0000_0008[WrLvOdtEn] and programming D18F2x[1,0]9C_x0000_0008[WrLvOdt]. BIOS programs D18F2x[1,0]9C_x0000_0008[WrLvOdt] with the D18F2x[1,0]F4_x182 value provided for writes to the rank targeted by training. See 2.9.3.7.1 [Write Levelization Training].

**Table 22: DIMM ODT pattern**

| DIMM0[1] | DIMM1[1] | D18F2x[1,0]F4_x180 | D18F2x[1,0]F4_x182 |
|---|---|---|---|
| SR[2] | - | 0000_0000h | 0000_0001h |
| DR[2] | - | 0000_0000h | 0000_0201h |
| - | SR | 0000_0000h | 0004_0000h |
| - | DR | 0000_0000h | 0804_0000h |
| SR/DR | SR/DR | 0101_0404h | 0905_0605h |

1. SR = Single rank, DR = Dual rank.
2. Only supported in systems which support a single DIMM per channel.

#### 2.9.3.4.6 DRAM Address Timing and Output Driver Compensation Control

This section describes the settings required for programming the timing on the address pins, the CS/ODT pins, and the CKE pins, as well as the processor ODT values controlled by D18F2x[1,0]9C_x0000_0000[ProcOdt]. Table 23 and Table 24 document the address timing, output driver settings, and processor ODT on a per channel basis for different DDR DIMM types. The DIMMs on each channel are numbered from 0 to n where DIMM0 is the DIMM closest to the processor on that channel and DIMMn is the DIMM farthest from the processor on that channel. DIMMs must be populated from farthest slot to closest slot to the processor on a per channel basis. Populations that are not shown in these tables are not supported. These tables document the optimal settings for motherboards which meet the relevant motherboard design guidelines.

**Table 23: BIOS recommendations for SO-DIMM address timings and output driver control**

| Condition | | | | | D18F2x[1,0]94 | D18F2x[1,0]9C_x0000_0004[1] | D18F2x[1,0]9C_x0000_0000 |
|---|---|---|---|---|---|---|---|
| NumDimmSlots | DdrRate | VDDIO | DIMM0 | DIMM1 | SlowAccessMode | | |
| 1 | 800 | 1.35, 1.5 | SR | - | 0 | 0000_0000h | 0000_2222h |
| 1 | 800 | 1.35, 1.5 | DR | - | 0 | 0000_0000h | 0000_2222h |
| 1 | 1066 | 1.35, 1.5 | SR | - | 0 | 003D_3D3Dh | 1000_2222h |
| 1 | 1066 | 1.35, 1.5 | DR | - | 0 | 0000_0000h | 1000_2222h |
| 1 | 1333 | 1.35, 1.5 | SR | - | 0 | 003D_3D3Dh | 2000_2222h |

**Table 23: BIOS recommendations for SO-DIMM address timings and output driver control**

| Condition | | | | | D18F2x[1,0]94 | D18F2x[1,0]9C_x0000_0004[1] | D18F2x[1,0]9C_x0000_0000 |
|---|---|---|---|---|---|---|---|
| NumDimmSlots | DdrRate | VDDIO | DIMM0 | DIMM1 | SlowAccessMode | | |
| 1 | 1333 | 1.35, 1.5 | DR | - | 0 | 0000_3D3Dh | 2000_2222h |
| 1 | 1600 | 1.5 | SR | - | 0 | 003C_3C3Ch | 3011_2222h |
| 1 | 1600 | 1.5 | DR | - | 1 | 0000_3C3Ch | 3011_2222h |
| 2 | 800 | 1.35, 1.5 | NP | SR | 0 | 0000_0000h | 0000_2222h |
| 2 | 800 | 1.35, 1.5 | NP | DR | 0 | 0000_0000h | 0000_2222h |
| 2 | 800 | 1.35, 1.5 | SR, DR | SR, DR | 1 | 0000_0039h | 2022_2323h |
| 2 | 1066 | 1.35, 1.5 | NP | SR | 0 | 003D_3D3Dh | 1000_2222h |
| 2 | 1066 | 1.35, 1.5 | NP | DR | 0 | 0000_0000h | 1000_2222h |
| 2 | 1066 | 1.35, 1.5 | SR, DR | SR, DR | 1 | 0000_0037h | 3022_2323h |
| 2 | 1333 | 1.5 | NP | SR | 0 | 003D_3D3Dh | 2011_2222h |
| 2 | 1333 | 1.5 | NP | DR | 0 | 0000_3D3Dh | 2011_2222h |
| 2 | 1333 | 1.5 | SR, DR | SR, DR | 1 | 0000_0035h | 3022_2323h |

1. The value of this register may be modified during Write Levelization Training. See 2.9.3.7.1 [Write Levelization Training].

**Table 24: BIOS recommendations for UDIMM address timings and output driver control**

| Condition | | | | | D18F2x[1,0]94 | D18F2x[1,0]9C_x0000_0004[1] | D18F2x[1,0]9C_x0000_0000 |
|---|---|---|---|---|---|---|---|
| NumDimmSlots | DdrRate | VDDIO | DIMM0 | DIMM1 | SlowAccessMode | | |
| 1 | 800 | 1.5 | SR | - | 0 | 0000_0000h | 0011_2222h |
| 1 | 800 | 1.5 | DR | - | 0 | 003B_0000h | 0011_2222h |
| 1 | 1066 | 1.5 | SR | - | 0 | 0000_0000h | 1011_2222h |
| 1 | 1066 | 1.5 | DR | - | 0 | 0038_0000h | 1011_2222h |
| 1 | 1333 | 1.5 | SR | - | 0 | 0000_0000h | 2011_2222h |
| 1 | 1333 | 1.5 | DR | - | 0 | 0036_0000h | 2011_2222h |
| 1 | 1600 | 1.5 | SR | - | 0 | 0000_0000h | 3011_2222h |

**Table 24: BIOS recommendations for UDIMM address timings and output driver control**

| Condition | | | | | D18F2x[1,0]94 | D18F2x[1,0]9C_x0000_0004[1] | D18F2x[1,0]9C_x0000_0000 |
|---|---|---|---|---|---|---|---|
| NumDimmSlots | DdrRate | VDDIO | DIMM0 | DIMM1 | SlowAccessMode | | |
| 1 | 1600 | 1.5 | DR | - | 1 | 0000_0000h | 3011_2222h |
| 1 | 1866 | 1.5 | SR | - | 0 | 0000_0000h | 3033_2222h |
| 1 | 1866 | 1.5 | DR | - | 1 | 0000_0000h | 3033_2222h |
| 2 | 800 | 1.5 | NP | SR | 0 | 0000_0000h | 0011_2222h |
| 2 | 800 | 1.5 | NP | DR | 0 | 003B_0000h | 0011_2222h |
| 2 | 800 | 1.5 | SR, DR | SR, DR | 0 | 0039_0039h | 2022_2322h |
| 2 | 1066 | 1.5 | NP | SR | 0 | 0000_0000h | 1011_2222h |
| 2 | 1066 | 1.5 | NP | DR | 0 | 0038_0000h | 1011_2222h |
| 2 | 1066 | 1.5 | SR, DR | SR, DR | 0 | 0035_0037h | 3022_2322h |
| 2 | 1333 | 1.5 | NP | SR | 0 | 0000_0000h | 2011_2222h |
| 2 | 1333 | 1.5 | NP | DR | 0 | 0036_0000h | 2011_2222h |
| 2 | 1333 | 1.5 | SR, DR | SR, DR | 1 | 0000_0035h | 3022_2322h |
| 2 | 1600 | 1.5 | NP | SR | 0 | 0000_0000h | 3011_2222h |
| 2 | 1600 | 1.5 | NP | DR | 1 | 0000_0000h | 3011_2222h |
| 2 | 1600 | 1.5 | SR, DR | SR, DR | 1 | 0000_0033h | 3022_2322h |

1. The value of this register may be modified during Write Levelization Training. See 2.9.3.7.1 [Write Levelization Training].

#### 2.9.3.4.7 NB P-states for DCT/Dram Initialization and Training

Before DRAM device initialization and training or prior to register restore when resuming the platform from S3, BIOS must force the processor to the NBP0 P-state. A subset of initialization and training must be repeated while forcing the processor to the NBP1 P-state. When DRAM training is complete, BIOS releases the force on the NB P-state. See 2.5.4.1.3 [Software Controlled NB P-states] and 2.9.3 [DCT/DRAM Initialization and Resume].

The following configuration registers contain multiple internal copies and must be programmed multiple times, once for each supported NB P-state. See D18F6x98[NbPsDbgEn, NbPsCsrAccSel] for information regarding how register context is selected.

- D18F2x[1,0]78[RdPtrInit].
- D18F2x[1,0]78[DisCutThroughMode].
- D18F2x[1,0]78[ForceCasToSlot].
- D18F2x[1,0]78[SlotSel].
- D18F2x[1,0]78[MaxSkipErrTrain].
- D18F2x[1,0]78[Slot1ExtraClkEn].

- D18F2x[1,0]78[MaxRdLatency]. See 2.9.3.7.5.1 [MaxRdLatency Training].
- D18F2x[1,0]F4_x30[DbeGskFifoNumerator].
- D18F2x[1,0]F4_x31[DbeGskFifoDenominator].
- D18F2x[1,0]F4_x32[DataTxFifoSchedDlySlot1].
- D18F2x[1,0]F4_x32[DataTxFifoSchedDlyNegSlot1].
- D18F2x[1,0]F4_x32[DataTxFifoSchedDlySlot0].
- D18F2x[1,0]F4_x32[DataTxFifoSchedDlyNegSlot0].

#### 2.9.3.5 DCT Training Specific Configuration

The DCT requires certain features be disabled during DRAM device initialization and training. BIOS should program the registers in Table 25 before and after DRAM device initialization and training. BIOS must quiesce all other forms of DRAM traffic on the channel being trained. See 2.9.3 [DCT/DRAM Initialization and Resume].

**Table 25: DCT training specific register values**

| Register | Value before | Value after |
|---|---|---|
| D18F2x[1,0]78[AddrCmdTriEn] | 0 | 1 |
| D18F2x[1,0]8C[DisAutoRefresh] | 1 | 0 |
| D18F2x[1,0]90[ForceAutoPchg] | 0 | 0 |
| D18F2x[1,0]90[DynPageCloseEn] | 0 | 0 |
| D18F2x[1,0]90[DbeSkidBufDis] | IF (D18F2x[1,0]F4_x40[Trcd] at target MEMCLK frequency > 0101b) THEN 0 ELSE 1 ENDIF. | |
| D18F2x[1,0]94[BankSwizzleMode] | 0 | 1 |
| D18F2x[1,0]94[PowerDownEn] | 0 | 1 |
| D18F2x[1,0]94[ZqcsInterval] | 00b | 10b |
| D18F2x[1,0]9C_x0000_000D[RxMaxDurDllNoLock] | 0h | See 2.9.3.9. |
| D18F2x[1,0]9C_x0000_000D[TxMaxDurDllNoLock] | 0h | See 2.9.3.9. |
| D18F2x[1,0]9C_x0D0F_0[F,7:0]10[EnRxPadStandby] | 0 | See 2.9.3.9. |
| D18F2xA4[ThrottleEn] | 00b | See 2.9.7 |
| D18F2x[1,0]A8[BankSwap] | 0 | 0 |
| D18F2x[1,0]A8[DbeGskMemClkAlignMode] | 00b | See 2.9.3 |
| D18F2x[1,0]F4_x06[TrdrdScEn] | 0 | 0 |
| D18F2x[1,0]F4_x32[DataTxFifoSchedDlyNegSlot1, DataTxFifoSchedDlySlot1, DataTxFifoSchedDlyNegSlot0, DataTxFifoSchedDlySlot0] | See 2.9.3.2.2. | See 2.9.3.2.2 |
| D18F2x110[DctSelIntLvEn] | 0 | See Table 30 |
| D18F2x11C[PrefCpuDis] | 1 | 0 |
| D18F2x11C[DctWrLimit] | 1Fh | 1Ch |
| D18F2x1C0[DramTrainPdbDis] | 0 | 1 |
| D18F3x7C | See Table 8 | See Table 8 |
| D18F3x17C | See Table 8 | See Table 8 |
| D18F3x188[EnCpuSerRdBehindNpIoWr] | 1 | 0 |

1. Programmed specific to the current memory configuration.

**Table 25: DCT training specific register values**

| Register | Value before | Value after |
|---|---|---|
| D18F4x1A8[DramSrEn] | 0 | 1 |
| MSRC001_0015[ForceUsRdWrSzPrb] | 0 | 0 |
| D18F6x78[DbeCmdThrottle] | 00h | 00h |
| 1. Programmed specific to the current memory configuration. | | |

#### 2.9.3.6 DRAM Device Initialization

BIOS initializes the DRAM devices and the controller using a software controlled sequence. See 2.9.3.6.1 [Software DDR3 Device Initialization].

DRAM initialization is complete after D18F2x[1,0]7C[EnDramInit] is written by BIOS from 1 to 0 in the software-controlled sequence.

See 2.9.3.5 [DCT Training Specific Configuration] for additional training requirements.

##### 2.9.3.6.1 Software DDR3 Device Initialization

BIOS should apply the following procedure to each DCT to initialize the DDR3 DIMMs on the channel. This procedure should be run only when booting from an unpowered state (ACPI S4, S5 or G3; not S3, suspend to RAM).

See 2.9.3.6.1.1 [DDR3 MR Initialization].

This procedure assumes the DRAM channel frequency is set and assumes D18F2x[1,0]7C[EnDramInit] is programmed to 1 prior to executing the following steps. See 2.9.3.2 and 2.9.3.2.2.

1. Configure the DCT registers, including MemClkFreq and MemClkFreqVal.
2. Program D18F2x[1,0]7C[EnDramInit] = 1.
3. Wait 200 us.
4. Program D18F2x[1,0]7C[DeassertMemRstX] = 1.
5. Wait 500 us.
6. Program D18F2x[1,0]7C[AssertCke] = 1.
7. Wait 360 ns.

The following steps are performed once for each channel:

8. Send MRS(2).
9. Send MRS(3). Ordinarily at this time, MrsAddress[2:0] = 000b.
10. Send MRS(1) with MrsAddress[7] = 0.
11. Send MRS(0) with MrsAddress[8] = 1.
12. Send a ZQCL command.
13. Program D18F2x[1,0]7C[EnDramInit] = 0.

BIOS instructs the DCT to send a ZQCL command by programming D18F2x[1,0]7C as follows:

1. Program MrsAddress[10] = 1.
2. Program SendZQCmd = 1.
3. Wait for SendZQCmd = 0.
4. Wait 512 MEMCLKs.

#### 2.9.3.6.1.1 DDR3 MR Initialization

BIOS instructs the DCT to send MRS commands by programming D18F2x[1,0]7C as follows:

1. Program MrsBank and MrsAddress as specified by Table 26, Table 27, Table 28, and Table 29:
   - MrsBank[2:0] = BA2:BA0.
   - MrsAddress[15:0] = A15:A0.
   - See D18F2x[1,0][4C:40][OnDimmMirror].
2. Program MrsChipSel as appropriate.
3. Program SendMrsCmd = 1.
4. Wait for SendMrsCmd = 0.

**Table 26. DDR3 MR0**

| Address Field | Field | Value |
|---|---|---|
| BA2:BA0 | MR Select | 000b |
| A15:A13 | Reserved | 000b |
| A12 | PPD | D18F2x[1,0]84[PchgPDModeSel] |
| A11:A9 | WR | D18F2x[1,0]84[Twr] |
| A8 | DLL | Controlled as required by the initialization sequence |
| A7 | TM | 0 |
| A6:A4,A2 | CAS Latency | {D18F2x[1,0]88[Tcl[2:0]], D18F2x[1,0]88[Tcl[3]]} |
| A3 | RBT | 1 |
| A1:A0 | BL | D18F2x[1,0]84[BurstCtrl] |

**Table 27. DDR3 MR1**

| Address Field | Field | Value |
|---|---|---|
| BA2:BA0 | MR Select | 001b |
| A15:A13 | Reserved | 000b |
| A12 | Qoff | 0b |
| A11 | TDQS | 0 |
| A10 | Reserved | 0b |
| A8 | Reserved | 0b |
| A7 | Level | Controlled as required by the initialization sequence |
| A4:A3 | AL | 00b |
| A9, A6, A2 | Rtt_Nom | See 2.9.3.4.5 [DRAM ODT Control] |
| A5, A1 | DIC | 01b |
| A0 | DLL | 0 |

**Table 28. DDR3 MR2**

| Address Field | Field | Value |
|---|---|---|
| BA2:BA0 | MR Select | 010b |
| A15:A11 | Reserved | 0_0000b |
| A10:A9 | Rtt_Wr | See 2.9.3.4.5 [DRAM ODT Control] |
| A8 | Reserved | 0b |
| A7 | SRT | See ASR |
| A6 | ASR | See DIMM SPD Byte 31: SDRAM Thermal and Refresh Options |
| A5:A3 | CWL | D18F2x[1,0]84[Tcwl] |
| A2:A0 | PASR | 000b |

**Table 29. DDR3 MR3**

| Address Field | Field | Value |
|---|---|---|
| BA2:BA0 | MR Select | 011b |
| A15:A3 | Reserved | 000_0000_0000b |
| A2 | MPR | 0b |
| A1:A0 | MPR Loc | 00b |

### 2.9.3.7 DRAM Training

This section describes the recommended methods used to train the processor DDR interface to DRAM for optimal functionality and performance. DRAM training is performed by BIOS after initializing the DRAM controller. See 2.9.3.6 [DRAM Device Initialization].

Some of the DRAM training steps described in this section require two passes if the target MEMCLK frequency is greater than the lowest supported MEMCLK frequency.For optimal software performance, software may defer the second pass (at target MEMCLK frequency) for each training step until after the first pass (at lowest supported frequency) of all other training steps are complete. See D18F2x[1,0]94[MemClkFreq].

See 2.9.3.5 [DCT Training Specific Configuration] for additional training requirements.

In the following subsections, lane is used to describe an 8-bit wide data group, each with its own timing control.

#### 2.9.3.7.1 Write Levelization Training

Write levelization involves using the phy to detect the edge of DQS with respect to the memory clock on the DIMM for write accesses to each lane.

Training is accomplished on a per channel, per DIMM basis. If the target frequency is greater than the lowest supported MEMCLK frequency then BIOS performs multiple passes; otherwise, only one pass is required. See 2.9.3.2.2.1 [Requirements for DRAM Frequency Change During Training].

- Pass 1: Configure the memory subsystem for the lowest supported MEMCLK frequency. See D18F2x[1,0]94[MemClkFreq].

- Pass 2 - Pass N: Configure the memory subsystem for the next higher supported MEMCLK frequency. Repeat until the target MEMCLK frequency is reached.

The following describes the steps used for each pass of write levelization training for each channel:

WLCriticalDelay = 0.
For each DIMM:

1. Prepare the DIMMs for write levelization using DDR3-defined MR commands. See 2.9.3.6.1.1 [DDR3 MR Initialization].
   A. Configure the output driver and on-die termination of the target DIMM as follows:
      - For the first rank of the target DIMM, enable write leveling mode and enable the output driver.
      - For all other ranks of the target DIMM, enable write leveling mode and disable the output driver.
      - For two or more DIMMs per channel, program Rtt_Nom of the target rank to the corresponding specified Rtt_Wr termination. Otherwise, configure Rtt_Nom of the target DIMM as normal. See 2.9.3.4.5 [DRAM ODT Control].
   B. Configure Rtt_Nom on the non-target DIMMs as normal. See 2.9.3.4.5.
2. Wait 40 MEMCLKs.
3. Configure the phy for write levelization training:
   A. Program D18F2x[1,0]9C_x0000_0008[WrtLvTrEn]=0.
   B. Program D18F2x[1,0]9C_x0000_0008[TrDimmSel] to specify the target DIMM to be trained.
   C. Program D18F2x[1,0]9C_x0000_0008[WrLvOdt[3:0]] to the proper ODT settings for the current memory subsystem configuration. See 2.9.3.4.5 and Table 22.
   D. Program D18F2x[1,0]9C_x0000_0008[WrLvOdtEn]=1.
   E. MFENCE.
   F. Wait 10 MEMCLKs to allow for ODT signal settling.
   G. For each lane program an initial value to registers D18F2x[1,0]9C_x0000_00[51:50] to set the gross and fine delay. See [The Write Levelization Seed Value] 2.9.3.7.1.1.
4. Perform write leveling of the devices on the DIMM:
   A. Program D18F2x[1,0]9C_x0000_0008[WrtLvTrEn]=1.
   B. MFENCE.
   C. Wait 200 MEMCLKs.
   D. Program D18F2x[1,0]9C_x0000_0008[WrtLvTrEn]=0.
   E. Read from registers D18F2x[1,0]9C_x0000_00[51:50] to get the gross and fine delay settings for the target DIMM and save these values.
5. Disable write levelization training so that the phy stops driving write levelization ODT.
   A. Program D18F2x[1,0]9C_x0000_0008[WrLvOdtEn]=0.
   B. MFENCE.
   C. Wait 10 MEMCLKs to allow for ODT signal settling.
6. Program the target DIMM back to normal operation by configuring the following (see step 1):
   - Configure all ranks of the target DIMM for normal operation.
   - Enable the output drivers of all ranks of the target DIMM.
   - For a two or more DIMM system, program the Rtt_Nom value for the target DIMM to the normal operating termination.
7. Calculate and program the final saved gross and fine delay values for each byte lane into D18F2x[1,0]9C_x0000_00[44:30][WrDqsGrossDly and WrDqsFineDly].
   A. (signed int) GrossDly = SeedGross + PhRecGrossDlyByte - SeedPreGross.
   B. IF (D18F2x[1,0]94[MemClkFreq] != target MEMCLK frequency) THEN
      IF (GrossDly < 0) THEN
      Program WrDqsFineDly = 0
      Program WrDqsGrossDly = 0
      ELSE

```
            Program WrDqsFineDly = PhRecFineDlyByte
            Program WrDqsGrossDly = GrossDly
        ENDIF.
    ELSE // target MEMCLK frequency
        Program WrDqsFineDly = PhRecFineDlyByte
        Program WrDqsGrossDly = GrossDly & 07h
        IF (GrossDly < 0) THEN
            // WLCriticalDelay is the unsigned difference between zero and the most negative
            // WrDqsDly total delay on this channel across all DIMMs and across all byte lanes.
            WLCriticalDelay = MAX((20h - WrDqsFineDly), WLCriticalDelay).
        ENDIF.
    ENDIF.
```

After write levelization training is completed for each channel:
IF (WLCriticalDelay > 0) THEN

1. NewClkFineDly = WLCriticalDelay.
2. Program D18F2x[1,0]9C_x0D0F_2[1:0]20[ClkFineDly]= NewClkFineDly.
3. Program D18F2x[1,0]9C_x0D0F_2[1:0]20[FenceBit]= (NewClkFineDly >= D18F2x[1,0]9C_x0000_000C[FenceThresholdTxPad]) ? 1 : 0.
4. Program D18F2x[1,0]9C_x0D0F_2[1:0]20[DllNukeLoad]= 1.
5. Program D18F2x[1,0]9C_x0D0F_2[1:0]20[DllNukeLoad]= 0.
6. Adjust DRAM address and command timing:
   A. AddrCmdTiming = D18F2x[1,0]9C_x0000_0004.
   B. AddrCmdTiming += (NewClkFineDly << 16) | (NewClkFineDly << 8) | NewClkFineDly.
   C. AddrCmdTiming &= 003F3F3Fh.
   D. Program D18F2x[1,0]9C_x0000_0004 = AddrCmdTiming.
7. Adjust WrDqs delays for all DIMMs and all byte lanes by adding NewClkFineDly to all the WrDqs total delays in D18F2x[1,0]9C_x0000_00[44:30]:
   A. WrDqsTiming = D18F2x[1,0]9C_x0000_00[44:30].
   B. WrDqsTiming += (NewClkFineDly << 16) | NewClkFineDly.
   C. WrDqsTiming &= 00FF00FFh.
   D. Program D18F2x[1,0]9C_x0000_00[44:30] = WrDqsTiming.

ENDIF.

#### 2.9.3.7.1.1 Write Levelization Seed Value

The seed value for pass 1 of write levelization training is design and platform specific and should be determined by characterization for best performance. The seed delay value must fall within +/- 1.20 ns, including PVT and jitter, of the measured clock delay.

The following steps are taken to determine the seed values needed to program the DRAM Phase Recovery Control Registers:

For each pass:

1. Calculate the total seed based on the following:
   - Pass 1: IF SO-DIMM THEN SeedTotal = 12h ELSE SeedTotal = 1Ah.
   - Pass 2 - Pass N:
     - SeedTotalPreScaling = the total delay values in D18F2x[1,0]9C_x0000_00[44:30] from the previous pass of write levelization training.
     - SeedTotal = FLOOR(SeedTotalPreScaling*(target frequency)/(frequency from previous pass)).

2. SeedGross = SeedTotal DIV 32.
3. SeedFine = SeedTotal MOD 32.
4. If (SeedGross is odd)
   then SeedPreGross = 1
   else SeedPreGross = 2.
5. Program D18F2x[1,0]9C_x0000_00[51:50][PhRecFineDlyByte] = SeedFine.
6. Program D18F2x[1,0]9C_x0000_00[51:50][PhRecGrossDlyByte] = SeedPreGross.

#### 2.9.3.7.2 DQS Receiver Enable Training

Receiver enable delay training is used to dynamically determine the optimal delay value for [The DRAM DQS Receiver Enable Timing Control] D18F2x[1,0]9C_x0000_00[24:10]. The optimal DQS receiver enable delay value is platform and load specific, and occurs in the middle of a received read preamble. The timing of the preamble includes the inbound DQS propagation delay, which is unknown by BIOS. The training for delay values involves:

1. Configuring the phy for an initial expected phase value (seed).
2. Generating a stream of read DQS edges from the DRAM by issuing multiple read commands.
3. The phy determining the phase between the received DQS edges and a reference clock.
4. Calculating a final delay value for enabling receivers during normal read operations using the phase determined by the phy.

Training is accomplished on a per channel, per DIMM, per rank basis. If the target frequency is greater than the lowest supported MEMCLK frequency then BIOS performs multiple passes; otherwise, only one pass is required. See 2.9.3.2.2.1 [Requirements for DRAM Frequency Change During Training].

- Pass 1: Configure the memory subsystem for the lowest supported MEMCLK frequency. See D18F2x[1,0]94[MemClkFreq].
- Pass 2 - Pass N: Configure the memory subsystem for the next higher supported MEMCLK frequency. Repeat until the target MEMCLK frequency is reached.

The following describes the steps used for each pass of receiver enable training for each channel:

Program D18F2x[1,0]78[MaxRdLatency] = 12h.

For each rank:

- Ensure that all ranks of the DIMM are configured for burst length 8 mode.

1. Program D18F2x[1,0]9C_x0000_0008[TrDimmSel] to specify the target DIMM to be trained.
2. For each lane program an initial value to registers D18F2x[1,0]9C_x0000_00[51:50] and D18F2x[1,0]9C_x0000_00[24:10] to set the gross and fine delay as specified in 2.9.3.7.2.1 [DQS Receiver Enable Training Seed Value].
3. Program D18F2x[1,0]9C_x0000_0008[DqsRcvTrEn]=1.
4. Issue 192 read requests to the target rank by issuing three sets of 64 read requests each. For each set of 64, the reads must be to consecutive dram column addresses (i.e. 64 bytes apart) and must not cross a naturally aligned 4 Kbyte boundary. To generate the needed continuous read streams for training, see 2.9.3.7.6 [DRAM Training Pattern Generation].
5. Program D18F2x[1,0]9C_x0000_0008[DqsRcvTrEn]=0.
6. Read D18F2x[1,0]9C_x0000_00[51:50][PhRecGrossDlyByte, PhRecFineDlyByte] to get the gross and fine delay values for each lane.
7. For each lane, calculate and program the corresponding receiver enable delay values for D18F2x[1,0]9C_x0000_00[24:10][DqsRcvEnGrossDelay, DqsRcvEnFineDelay]. Save the result for use

later.
  - DqsRcvEnFineDelay = PhRecFineDlyByte
  - DqsRcvEnGrossDelay = SeedGross + PhRecGrossDlyByte - SeedPreGross +1.
- For each rank pair on a dual-rank DIMM, compute the average value of the total delays saved during the training of each rank and program the result in D18F2x[1,0]9C_x0000_00[24:10][DqsRcvEnGrossDelay, DqsRcvEnFineDelay].

#### 2.9.3.7.2.1 DQS Receiver Enable Training Seed Value

The seed value for pass 1 of receiver enable delay training is design and platform specific and should be determined by characterization for best performance. The seed value represents the total delay from a reference point to the left edge of the read preamble on a read CAS measured at the processor pins, in 1 UI/32 increments. The reference point is defined as the clock in which CAS is asserted + CL - 1. This value is expected to be larger than 2 UI in the steps below. The phy adds a fixed offset to the delay seed value prior to sampling read DQS edges.

The following steps are taken to determine the seed values needed to program the DRAM Phase Recovery Control Registers:

For each pass and each lane:

1. Calculate the total seed based on the following:
   - Pass 1: SeedTotal = 3Bh + the total delay values obtained from the first pass of write levelization training. See 2.9.3.7.1 [Write Levelization Training].
   - Pass 2 - Pass N:
     - SeedTotalPreScaling = (the total delay values in D18F2x[1,0]9C_x0000_00[24:10] from the previous pass of DQS receiver enable training) - 20h.
     - SeedTotal = FLOOR(SeedTotalPreScaling*(target frequency)/(frequency from previous pass)).
2. SeedGross = SeedTotal DIV 32.
3. SeedFine = SeedTotal MOD 32.
4. If (SeedGross is odd)
   then SeedPreGross = 1
   else SeedPreGross = 2.
5. Program D18F2x[1,0]9C_x0000_00[51:50][PhRecFineDlyByte] = SeedFine.
6. Program D18F2x[1,0]9C_x0000_00[51:50][PhRecGrossDlyByte] = SeedPreGross.
7. Program D18F2x[1,0]9C_x0000_00[24:10][DqsRcvEnGrossDelay] = SeedGross.

#### 2.9.3.7.3 DQS Receiver Enable Cycle Training

Receiver enable delay cycle training is used to train the gross delay settings of [The DRAM DQS Receiver Enable Timing Control] D18F2x[1,0]9C_x0000_00[24:10] to the middle of the received read preamble using the phy phase results.

For each rank and lane:
1. Program D18F2x[1,0]9C_x0D0F_0[F,7:0]30[BlockRxDqsLock] = 1.
2. RxEnOrig = D18F2x[1,0]9C_x0000_00[24:10][DqsRcvEnGrossDelay, DqsRcvEnFineDelay] result from 2.9.3.7.2 [DQS Receiver Enable Training].
3. RxEnOffset = MOD(RxEnOrig + 0x10, 0x40)
4. For each DqsRcvEn value beginning from RxEnOffset incrementing by 1 MEMCLK:
   A. Program D18F2x[1,0]9C_x0000_00[24:10][DqsRcvEnGrossDelay, DqsRcvEnFineDelay] with the current value.

B. Initialize Receive FIFO pointers:
   a. Program D18F2x[1,0]78[RxPtrInitReq]=1.
   b. Wait for D18F2x[1,0]78[RxPtrInitReq]=0.
C. Perform 2.9.3.7.4 [DQS Position Training].
   • Record the result for the current DqsRcvEn setting as a pass or fail depending if a data eye is found.

5. Process the array of results and determine a pass-to-fail transition.
   A. DqsRcvEnCycle = the total delay value of the pass result.
   B. Program D18F2x[1,0]9C_x0000_00[24:10][DqsRcvEnGrossDelay, DqsRcvEnFineDelay] = DqsRcvEnCycle - 0x10.
   C. Initialize Receive FIFO pointers:
      a. Program D18F2x[1,0]78[RxPtrInitReq]=1.
      b. Wait for D18F2x[1,0]78[RxPtrInitReq]=0.
6. Program D18F2x[1,0]9C_x0D0F_0[F,7:0]30[BlockRxDqsLock] = 0.

#### 2.9.3.7.4 DQS Position Training

DQS position training is used to place the DQS strobe in the center of the read DQ data eye and to center the write DQ data eye across the write DQS strobe. Determining the correct DRAM DQS and DQ delay settings for both reads and writes is done by performing a two dimensional search of the delay settings found in [The DRAM Read DQS Timing Control] D18F2x[1,0]9C_x0000_0[1:0]0[6:5] and [The DRAM Write Data Timing] D18F2x[1,0]9C_x0000_0[1:0]0[2:1].

Training is accomplished on a per channel, per rank, and per lane basis. BIOS uses the mutual passing delay values of each rank of a dual rank DIMM to calculate the optimal delay values.

For DQS position training, BIOS generates a training pattern using continuous read or write data streams. See 2.9.3.7.6 [DRAM Training Pattern Generation]. A 256-bit-time training pattern is recommended for optimal results.

Prior to DQS position training, BIOS must program D18F2x[1,0]78[MaxRdLatency] based on the current greatest value of D18F2x[1,0]9C_x0000_00[24:10]. See 2.9.3.7.5 [Calculating MaxRdLatency].

The following describes the steps used for DQS position training for each channel:

For each rank and lane:

1. Select a 64 byte aligned test address.
2. For each write data delay value in D18F2x[1,0]9C_x0000_0[1:0]0[2:1] from Wr-DQS to Wr-DQS plus 1 UI, using the Wr-DQS delay value found in 2.9.3.7.1 [Write Levelization Training]:
   A. Program the write data delay value for the current lane.
   B. Write the DRAM training pattern to the test address.
   C. For each read DQS delay value in D18F2x[1,0]9C_x0000_0[1:0]0[6:5] from 0 to 1 UI:
      a. Program the read DQS delay value for the current lane.
      b. Read the DRAM training pattern from the test address.
      c. Record the result for the current settings as a pass or fail depending if the pattern is read correctly.
3. Process the array of results and determine the longest string of consecutive passing read DQS delay values.
   • If the read DQS delay results for the current lane contain three or more consecutive passing delay values, then program D18F2x[1,0]9C_x0000_0[1:0]0[6:5] with the average value of the smallest and largest delay values in the string of consecutive passing results.
4. Process the array of results and determine the longest string of consecutive passing write data delay values for the read DQS delay value found in the step above.
   • If the write data delay results for the current lane contain three or more consecutive passing delay values, then program D18F2x[1,0]9C_x0000_0[1:0]0[2:1] with the average value of the smallest and largest

delay values in the string of consecutive passing results.
See Figure 6.

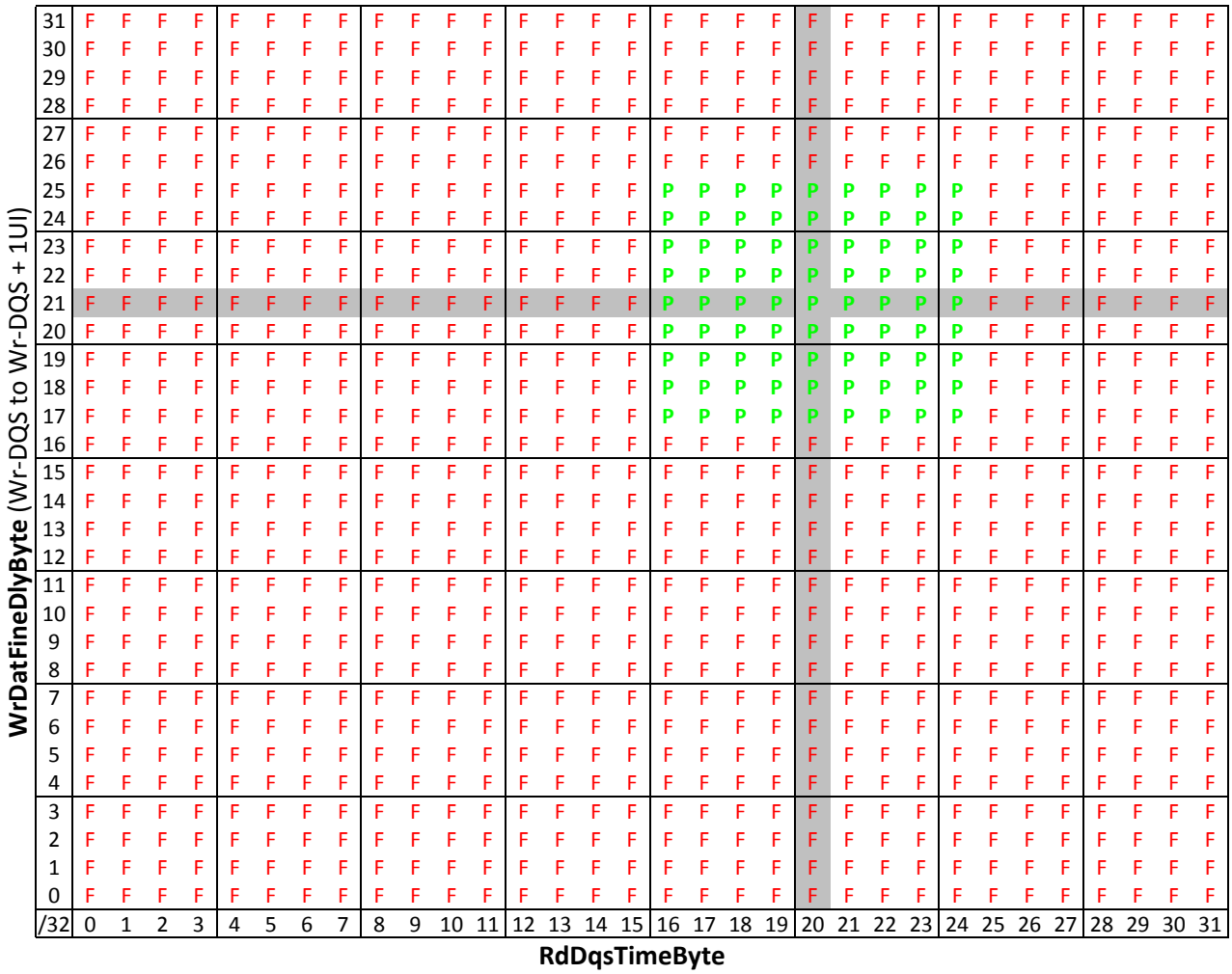


**Figure 6: DQS Position Training Example Results**

In some cases, a non-zero process, voltage, and temperature dependent insertion delay is added to the DLL programmed read DQS delay. This has the effect of sampling data later than intended and can result in missing the left edge of the passing region when sweeping from 0 to 1 UI because a read DQS delay value of 0 is already in the passing region. Since DQS is periodic, BIOS can recover the missing information by adjusting the algorithm described above to analyze both the in phase data and the data shifted by one bit time at each step of the read DQS delay sweep. See D18F2x1E8[TrainCmpSts2, TrainCmpSts].

As shown in Figure 7, for each delay setting BIOS records a passing result of $P_?$ for the data comparison shifted by one bit time if the data at bit times N=0, 1, …, 6, is read correctly when compared against the data written at bit times N=1, 2, …, 7. In the array of results, these passing values make up the left piece of information that had been lost due to insertion delay. In order to process the array of results, BIOS calculates the read DQS delay value for a $P_?$ result as RdDqsTimeByte minus 1 UI.

| WrDatFineDlyByte (Wr-DQS to Wr-DQS + 1UI) | RdDqsTimeByte | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 31 | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F |
| 30 | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F |
| 29 | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F |
| 28 | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F |
| 27 | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F |
| 26 | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F |
| 25 | P | P | P | P | P | P | P | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | P? | P? |
| 24 | P | P | P | P | P | P | P | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | P? | P? |
| 23 | P | P | P | P | P | P | P | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | P? | P? |
| 22 | P | P | P | P | P | P | P | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | P? | P? |
| 21 | P | P | P | P | P | P | P | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | P? | P? |
| 20 | P | P | P | P | P | P | P | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | P? | P? |
| 19 | P | P | P | P | P | P | P | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | P? | P? |
| 18 | P | P | P | P | P | P | P | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | P? | P? |
| 17 | P | P | P | P | P | P | P | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | P? | P? |
| 16 | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F |
| 15 | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F |
| 14 | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F |
| 13 | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F |
| 12 | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F |
| 11 | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F |
| 10 | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F |
| 9 | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F |
| 8 | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F |
| 7 | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F |
| 6 | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F |
| 5 | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F |
| 4 | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F |
| 3 | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F |
| 2 | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F |
| 1 | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F |
| 0 | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F |
| /32 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| ? | -32 | -31 | -30 | -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 |

**Figure 7: DQS Position Training Insertion Delay Recovery Example Results**

#### 2.9.3.7.5 Calculating MaxRdLatency

The MaxRdLatency value determines when the processor can receive incoming data from the DCT. Calculating MaxRdLatency consists of summing all the synchronous and asynchronous delays in the path from the processor to the DRAM and back at a given MEMCLK frequency. BIOS incrementally calculates the MaxRdLatency and then finally programs the value into D18F2x[1,0]78[MaxRdLatency].

The following steps describe the algorithm used to compute D18F2x[1,0]78[MaxRdLatency] used for DRAM training. P, N, and T are used as temporary placeholders for the incrementally summed value.

P = N = T = 0.

1. Pre- Tx FIFO adjustments
   A. IF (D18F2x[1,0]9C_x0000_0004[AddrCmdSetup]!= D18F2x[1,0]9C_x0000_0004[CkeSetup]) THEN P = P + 1
   B. IF (D18F2x[1,0]A8[DbeGskMemClkAlignMode]==01b || (D18F2x[1,0]A8[DbeGskMemClkAlignMode]==00b && !(D18F2x[1,0]9C_x0000_0004[AddrCmdSetup]== D18F2x[1,0]9C_x0000_0004[CsOdtSetup]==

D18F2x[1,0]9C_x0000_0004[CkeSetup])))
THEN P = P + 1

C. IF (D18F2x[1,0]94[SlowAccessMode]==1)
THEN P = P + 2

2. P = P + PtrSeparation
   - PtrSeparation = ((16 + RdPtrInitMin - D18F2x[1,0]78[RdPtrInit]) MOD 16)
   - RdPtrInitMin = 7.
3. P = P + 2
4. If (D18F2x[1,0]9C_x0000_0004[AddrCmdSetup]==0 && D18F2x[1,0]9C_x0000_0004[CsOdtSetup]==0 && D18F2x[1,0]9C_x0000_0004[CkeSetup]==0)
   then P = P + 1
   else P = P + 2
5. P = P + (2 * (D18F2x[1,0]88[Tcl] clocks - 1))
6. P = P + CEIL(MAX (total delay in D18F2x[1,0]9C_x0000_00[24:10] + D18F2x[1,0]9C_x0000_0[1:0]0[6:5][RdDqsTime]))
   - Use maximum DqsRcvEn total delay plus RdDqsTime across all DIMMs and all byte lanes.
   - Prior to DQS position training, use maximum value for RdDqsTime.
7. If (D18F2x[1,0]78[DisCutThroughMode]==0)
   then P = P + 3
   else P = P + 7
8. P = P + 7.5
9. T = T + 2586 ps
10. N = (P/(MemClkFreq * 2) + T) * NclkFreq
    - NclkFreq = NCLK frequency as defined by D18F3xDC[NbPs0NclkDiv] or D18F6x90[NbPs1NclkDiv] for given NB P-state.
    - MemClkFreq = MEMCLK frequency as defined by D18F2x[1,0]94[MemClkFreq].
11. D18F2x[1,0]78[MaxRdLatency] = CEIL(N)

#### 2.9.3.7.5.1 MaxRdLatency Training

After DRAM training, BIOS optimizes D18F2x[1,0]78[MaxRdLatency] using the following algorithm:

For MaxRdLatency training, BIOS generates a training pattern using continuous read or write data streams. See 2.9.3.7.6 [DRAM Training Pattern Generation]. The following three cache line pattern is used to train the MaxRdLatency value:

```
0C3C_FF52_6E0E_3FACh
49C5_B613_4A68_8181h
5C16_50E3_7C78_0BA6h
0C67_53E6_0C4F_9D76h
BABF_B6CA_2055_35A5h
0C5F_1C87_610E_6E5Fh
14C9_C383_4884_93CEh
9CE8_F615_F5B9_A5CDh

C38F_1B4C_AAD7_14B5h
669F_7562_72ED_647Ch
4A89_8B30_5233_F802h
3326_B465_10A4_0617h
C807_E3D3_5538_6E04h
14B4_E63A_AB49_E193h
EA51_7C45_67DF_2495h
```

```
F814_0C51_7624_CE51h

B61D_D0C9_4824_BD23h
E8F3_807D_072B_CFBEh
25E3_0C47_919E_A373h
4DA8_0A5A_FEB1_2958h
792B_0076_E9A0_DDF8h
F025_B496_E81C_73DCh
8085_94FE_1DB7_E627h
655C_7783_8266_8268h
```

For each channel:

1. IF (D18F2x[1,0]88[Tcl] <= 2) THEN program D18F2x[1,0]78[MaxSkipErrTrain]=0 ELSE program D18F2x[1,0]78[MaxSkipErrTrain]=1 ENDIF.
2. Program D18F2x[1,0]78[SlotSel]=0.
3. Program D18F2x[1,0]78[ForceCasToSlot]=1.
4. Calculate a starting MaxRdLatency delay value by executing the steps in section 2.9.3.7.5. Save this as CalcMaxRdLatency.
5. Select six linearly contiguous 64 byte aligned test addresses associated with the DIMM that has the worst case D18F2x[1,0]9C_x0000_00[24:10] register setting.
6. Write the six DIMM test addresses by twice repeating the three cache line pattern given above.
7. Program D18F2x[1,0]78[MaxRdLatency] = CalcMaxRdLatency.
8. For (i = 0; i < 100; i++):
   A. Read the DIMM test addresses.
   B. Compare the values read against the pattern written.
      - If pattern is read correctly then save current MaxRdLatency as TrainedMaxRdLatency and decrement D18F2x[1,0]78[MaxRdLatency].
9. Perform slot1 specific training:
   A. Program D18F2x[1,0]78[SlotSel]=1.
   B. Program D18F2x[1,0]78[MaxRdLatency] = TrainedMaxRdLatency. For (i = 0; i < 100; i++):
      a. Read the DIMM test addresses.
      b. Compare the values read against the pattern written.
      c. If pattern is read correctly then program D18F2x[1,0]78[SlotExtraClkEn]=0 and jump to step 10.
      d. If pattern is not read correctly and i = 99 then program D18F2x[1,0]78[SlotExtraClkEn]=1.
10. Program D18F2x[1,0]78[MaxRdLatency] = TrainedMaxRdLatency + TrainingOffset.
    - IF (D18F2x[1,0]78[MaxSkipErrTrain]==0 && NCLK!=MEMCLK && NCLK!=MEMCLK/2) THEN TrainingOffset = 3
      ELSE TrainingOffset = 2 ENDIF.
11. Program D18F2x[1,0]78[MaxSkipErrTrain]=0.
12. Program D18F2x[1,0]78[ForceCasToSlot]=0.

#### 2.9.3.7.6 DRAM Training Pattern Generation

DRAM training relies on generating a sequence of reads or writes between the processor and DRAM such that worst case electrical interactions are created. This section describes how these sequences are generated. DRAM training pattern generation uses the read/write training buffer for data storage. During write pattern generation, the write training buffer is filled by software, and the contents are burst to the DRAM interface. Conversely for reads, data bursts from the DRAM interface are stored in the read training buffer. When read data bursts are stored in the read training buffer they are automatically compared against the data in the write training buffer. Software accesses the compare results by reading D18F2x1E8[TrainCmpSts2,TrainCmpSts].

Two address modes are available for DRAM training pattern generation. For generating a continuous sequence

of reads or writes to the same rank, continuous pattern generation mode is used. See 2.9.3.7.6.1 [Continuous Pattern Generation]. To generate sequences of accesses to up to four different ranks or DIMMs, alternative address mode is used. See 2.9.3.7.6.2 [Alternative Address Mode].

#### 2.9.3.7.6.1 Continuous Pattern Generation

Continuous pattern generation mode uses a single linearly contiguous sequence of DRAM training addresses to read or write training patterns to a single rank. Addresses must not cross a naturally aligned 4 Kbyte boundary.

Program the following to enable continuous pattern generation mode:

1. Program D18F2x1C0[AltAddrEn]=0.
2. Program {D18F2x1CC[TrainAddrPtr[39:38]], D18F2x1C8[TrainAddrPtr[37:6]]} to the DRAM training start address.

TrainAddrPtr[39:6] is incremented by hardware after every cache line transfer.

#### 2.9.3.7.6.2 Alternative Address Mode

Alternative address mode uses multiple linearly contiguous sequences of DRAM training addresses to read or write training patterns to different DIMMs or ranks. In this mode, the address alternates between four address pointers, and the DRAM data stream may not be continuous across multiple address pointers. Each address pointer consists of a pointer field and an iteration field. When a training sequence is initiated (D18F2x1C0[RdTrainGo or WrTrainGo]=1) and alternative address mode is enabled (D18F2x1C0[AltAddrEn]=1), hardware sequences through the training address pointers transferring the number of cache lines specified for each pointer (TrainAddrPtrIt +1) until the number of cache lines specified by D18F2x1C0[TrainLength] has been transferred. The associated DRAM training address pointer is incremented by hardware after every cache line transfer.

Program the following to enable alternative address mode pattern generation:

1. Program D18F2x1C0[AltAddrEn]=1.
2. Program {D18F2x1CC[TrainAddrPtr[39:38]], D18F2x1C8[TrainAddrPtr[37:6]]} to the DRAM training start address.
3. Program {D18F2x1CC[AltAddr1Ptr[39:38]], D18F2x[1E0:1D8][AltAddr1Ptr[37:6]]}.
4. Program {D18F2x1CC[AltAddr2Ptr[39:38]], D18F2x[1E0:1D8][AltAddr2Ptr[37:6]]}.
5. Program {D18F2x1CC[AltAddr3Ptr[39:38]], D18F2x[1E0:1D8][AltAddr3Ptr[37:6]]}.

#### 2.9.3.7.6.3 Read Pattern Generation

Perform the following steps to read a training pattern from DRAM:

The DCT requires certain features be disabled to achieve continuous patterns. See 2.9.3.5 [DCT Training Specific Configuration].

1. Program D18F2x1C0[RdDramTrainMode]=1.
2. Program D18F2x1C0[TrainLength] to the appropriate number of cache lines.
3. Program the DRAM training address as follows:
   - If continuous pattern generation mode is desired, see 2.9.3.7.6.1 [Continuous Pattern Generation].
   - Else for alternative address mode, see 2.9.3.7.6.2 [Alternative Address Mode].
4. Program D18F2x1D0[WrTrainBufAddr]=000h.
5. Program D18F2x1C0[RdTrainGo]=1.

6. Wait for D18F2x1C0[RdTrainGo]=0.
7. Read D18F2x1E8[TrainCmpSts] and D18F2x1E8[TrainCmpSts2].
8. Program D18F2x1C0[RdDramTrainMode]=0.

#### 2.9.3.7.6.4 Write Pattern Generation

Write DRAM training is accomplished using the write training buffer. Perform the following steps to initialize the write training buffer:

1. Program D18F2x1C0[WrDramTrainMode]=1.
2. Program D18F2x1C0[TrainLength] to the appropriate number of cache lines.
3. Program D18F2x1D0[WrTrainBufAddr]=000h. Successively write each dword of the training pattern to D18F2x1D4.

After initializing the write training buffer, perform the following steps to write the pattern to DRAM:

The DCT requires certain features be disabled to achieve continuous patterns. See 2.9.3.5 [DCT Training Specific Configuration].

4. Program D18F2x1D0[WrTrainBufAddr]=000h.
5. Program the DRAM training address as follows:
   - If continuous pattern generation mode is desired, see 2.9.3.7.6.1 [Continuous Pattern Generation].
   - Else for alternative address mode, see 2.9.3.7.6.2 [Alternative Address Mode].
6. Program D18F2x1C0[WrTrainGo]=1.
7. Wait for D18F2x1C0[WrTrainGo]=0.
8. Program D18F2x1C0[WrDramTrainMode]=0.
9. If training is not complete, program D18F2x1C0[WrDramTrainMode]=1 and go to step 4 to issue next set of training writes.

### 2.9.3.8 DRAM Channel Disable

The following steps are performed to disable an unused DRAM channel:

For the channel to be disabled:
1. Program D18F2x[1,0]9C_x0000_000C[CKETri] = 11b.
2. Wait 24 MEMCLKs.
3. Program D18F2x[1,0]94[DisDramInterface] = 1.
4. Program D18F2x[1,0]9C_x0000_000B = 8080_0000h.

### 2.9.3.9 DRAM Phy Power Savings

To configure the phy for lower power consumption, the following steps are performed on each enabled DRAM channel:

1. Program D18F2x[1,0]88[MemClkDis] to disable unused MEMCLK pins.
2. Program D18F2x[1,0]9C_x0D0F_2[1:0]30[PwrDn] = 1 for unused MEMCLK pairs.
3. Program D18F2x[1,0]9C_x0000_000C[CKETri, ODTTri, ChipSelTri] to disable unused pins.
4. Program D18F2x[1,0]9C_x0D0F_0[F,7:0]13[DllDisEarlyU] = 1b.
5. Program D18F2x[1,0]9C_x0D0F_0[F,7:0]13[DllDisEarlyL] = 1b.
6. Program D18F2x[1,0]9C_x0D0F_0[F,7:0]13[RxDqsUDllPowerDown] = 1b.
7. Program D18F2x[1,0]9C_x0D0F_812F[7, 5, 0] = {1b, 1b, 1b}.

8. IF ((DimmsPopulated == 1) && ((D18F2x[1,0]9C_x0000_0000[CkeDrvStren]==3'b010) || (D18F2x[1,0]9C_x0000_0000[CkeDrvStren]==3'b011))) THEN program D18F2x[1,0]9C_x0D0F_C000[LowPowerDrvStrengthEn]= 1 ELSE program D18F2x[1,0]9C_x0D0F_C000[LowPowerDrvStrengthEn]= 0 ENDIF.
9. IF (D18F2x[1,0]94[MemClkFreq] <= 800 MHz) THEN program D18F2x[1,0]9C_x0D0F_0[F,7:0]10[EnRxPadStandby] = 1 ELSE program D18F2x[1,0]9C_x0D0F_0[F,7:0]10[EnRxPadStandby] = 0 ENDIF.
10. Program D18F2x[1,0]9C_x0000_000D as follows:
    - TxMaxDurDllNoLock/RxMaxDurDllNoLock = 7h.
    - TxCPUpdPeriod/RxCPUpdPeriod = 011b.
    - TxDLLWakeupTime/RxDLLWakeupTime = 11b.

### 2.9.4 Memory Interleaving Modes

The processor supports two different types of interleaving modes:
- Chip select: interleaving between the DIMM ranks of a channel. This mode interleaves the physical address space over multiple DIMM ranks, as opposed to each DIMM owning a single consecutive contiguous address space. This is accomplished by using lower-order address bits to select between DIMMs. See D18F2x[1,0][4C:40].
- Channel: interleaving between the two 64-bit channels of a processor. This mode interleaves the physical address space over two channels, as opposed to each channel owning a single consecutive contiguous address space. This is accomplished by using lower-order address bits to select between channels. See D18F2x110[DctSelIntLvEn].

Any combination of these interleaving modes may be enabled concurrently. In addition, it is possible to remap the physical addresses corresponding to an interleaved region of memory. See Table 30 and D18F2x10C.

**Table 30. Recommended interleave configurations**

| Interleaving Mode | Enabled | Disabled |
|---|---|---|
| Chip Select Interleaving | Number of chip selects installed on the channel is power of two. | Number of chip selects installed on the channel is not power of two. |
| Channel Interleaving | DIMMs are present on both channels. | DIMMs are not present on both channels. |
| Interleave Region Remapping[1] | DIMMs are present on both channels but the channels do not have the same amount of DRAM. | Both channels have the same amount of DRAMor DIMMs are not present on both channels. |
| 1. The channel interleave region should always include the frame buffer. | | |

#### 2.9.4.1 Chip Select Interleaving

The chip select memory interleaving mode requires that all interleaved chip selects are of the same bank addressing type, and that the number of interleaved chip selects is a power of two. A BIOS algorithm for programming [The DRAM CS Base Address Registers] D18F2x[1,0][4C:40] and [The DRAM CS Mask Register] D18F2x[1,0][64:60] in memory interleaving mode is as follows:

1. Program all DRAM CS Base Address and DRAM CS Mask registers using contiguous normalized address mapping.
2. For each enabled chip select, swap the corresponding BaseAddr[36:27] bits with the BaseAddr[21:13] bits

as defined in Table 31.

3. For each enabled chip select, swap the corresponding AddrMask[36:27] bits with the AddrMask[21:13] bits as defined in Table 31.

**Table 31. DDR3 swapped normalized address lines for CS interleaving**

| DIMM Address Map[1] | Chip Select Size | Swapped Base Address and Address Mask bits | |
|---|---|---|---|
| | | 4 way CS interleaving | 2 way CS interleaving |
| 0001b | 256-MB | [29:28] and [17:16] | [28] and [16] |
| 0010b | 512-MB | [30:29] and [17:16] | [29] and [16] |
| 0101b | 1-GB | [31:30] and [17:16] | [30] and [16] |
| 0111b | 2-GB | [32:31] and [17:16] | [31] and [16] |
| 1010b | 4-GB | [33:32] and [17:16] | [32] and [16] |
| 1011b | 8-GB | [34:33] and [18:17] | [33] and [17] |

1. See [The DRAM Bank Address Mapping Register] D18F2x[1,0]80.

The following is an example of interleaving a 64-bit interface to DDR3 DRAM. The DRAM memory consists of one 512 Mbyte dual rank DDR3 DIMM.

1. The register settings for contiguous memory mapping are:
   D18F2x80 = 0000_0001h // CS0/1 = 256 MB
   D18F2x40 = 0000_0001h // 0 MB base
   D18F2x44 = 0010_0001h // 256 MB base = 0 MB + 256 MB
   D18F2x60 = 0008_3FE0h // CS0/CS1 = 256 MB
2. The BaseAddr bits to be swapped are defined in Table 31: 256MB chip select size row, 2 way CS interleaving column. The BaseAddr[28] bit is defined with D18F2x[1,0][4C:40][20]. The BaseAddr[16] bit is defined with D18F2x[1,0][4C:40][8].
   D18F2x40 = 0000_0001h
   D18F2x44 = 0000_0101h
3. The AddrMask bits to be swapped are the same as the BaseAddr bits defined in the previous step. The AddrMask[28] bit is defined with D18F2x[1,0][64:60][20]. The AddrMask[16] bit is defined with D18F2x[1,0][64:60][8].
   D18F2x60 = 0018_3EE0h

### 2.9.5 Memory Hoisting

Memory hoisting is defined as reclaiming (relocating) the unusable DRAM space that would naturally reside in the MMIO hole just below the 4GB address level. This memory is repositioned above the 4GB level when the registers that control memory hoisting, [The DRAM Hole Address Register] D18F1xF0, [The DRAM Controller Select Low Register] D18F2x110, and [The DRAM Controller Select High Register] D18F2x114, are properly configured.

The region of DRAM that is hoisted is defined to be from D18F1xF0[DramHoleBase] to the 4GB level. The beginning of the upper memory space, as specified by D18F2x110[DctSelHi, DctSelBaseAddr], is allowed to occur at any relationship with respect to the MMIO hole. This applies to both non-interleaved and interleaved configurations.

The memory hoisting offset fields, D18F1xF0[DramHoleOffset] and D18F2x114[DctSelBaseOffset], are pro-

grammed based on the following parameters:
- D18F1xF0[DramHoleBase] specifies the base address of the MMIO hole below the 4G level.
- D18F2x110[DctSelBaseAddr] specifies the base address of the upper memory space owned by the DCT specified by D18F2x110[DctSelHi].
- D18F2x110[DctSelIntLvEn] specifies if interleaving between the two DCTs is enabled (channel interleave mode).
- D18F2x[1,0][4C:40][CSEnable] specifies if one or both DCTs are enabled.

DramHoleSize is defined in order to simplify the following equations in this section and is calculated as follows: DramHoleSize[31:24] = (100h-DramHoleBase[31:24]).

#### 2.9.5.1 DctSelBaseAddr Programming

D18F2x110[DctSelBaseAddr], for both interleaved and non-interleaved, is adjusted for hoisting based on D18F1xF0[DramHoleBase] and the hoisting disabled value of DctSelBaseAddr, called (non-hoisted-DctSelBaseAddr). For the remainder of this section DctSelBaseAddr refers to the hoisting enabled value of DctSelBaseAddr.

DctSelBaseAddr is programmed to one of the following equations based on the scenario:
- **Case 1**: for any of the following conditions:
  - non-hoisted-DctSelBaseAddr >= DramHoleBase. (Non-hoisted-DctSelBaseAddr[39:27] >= {00h,DramHoleBase[31:27]})

  then:
  - DramHoleBase[26:24] must be 000b.
  - DctSelBaseAddr[39:27] = Non-hoisted-DctSelBaseAddr[39:27] + {00h,DramHoleSize[31:27]}.
- **Case 2**: for any of the following conditions:
  - non-hoisted-DctSelBaseAddr < DramHoleBase. (Non-hoisted-DctSelBaseAddr[39:27] < {00h,DramHoleBase[31:27]})

  then:
  - DctSelBaseAddr is unchanged.

#### 2.9.5.2 DramHoleOffset Programming

D18F1xF0[DramHoleOffset] is programmed to one of the following equations based on the scenario:
- **Case 1**: for any of the following conditions:
  - 1 DCT is enabled
  - 2 DCTs are enabled in non-interleaved mode and (non-hoisted-DctSelBaseAddr > DramHoleBase)
  - 2 DCTs are enabled in interleaved mode and the same size (D18F2x110[DctSelHiRngEn]=0)
  - 2 DCTs are enabled in interleaved mode and (non-hoisted-DctSelBaseAddr > DramHoleBase)

  then:
  - DramHoleOffset[31:23] = {DramHoleSize[31:24],0b};
- **Case 2**: for any of the following conditions:
  - 2 DCTs are enabled in non-interleaved mode and (non-hoisted-DctSelBaseAddr < DramHoleBase)

  then:
  - DramHoleOffset[31:23] = {DramHoleSize[31:24],0b}+{DctSelBaseAddr[31:27],0000b};
- **Case 3**: for any of the following conditions:
  - 2 DCTs are enabled in interleaved mode and (non-hoisted-DctSelBaseAddress < DramHoleBase)

  then:
  - DramHoleOffset[31:23] = {DramHoleSize[31:24],0b}+{0b,DctSelBaseAddr[31:27],000b};
- **Case 4**: for any of the following conditions:
  - 2 DCTs are enabled in interleaved or non-interleaved mode and (non-hoisted-DctSelBaseAddr = Dram-

HoleBase)

then:

- DramHoleOffset[31:23] is not used.

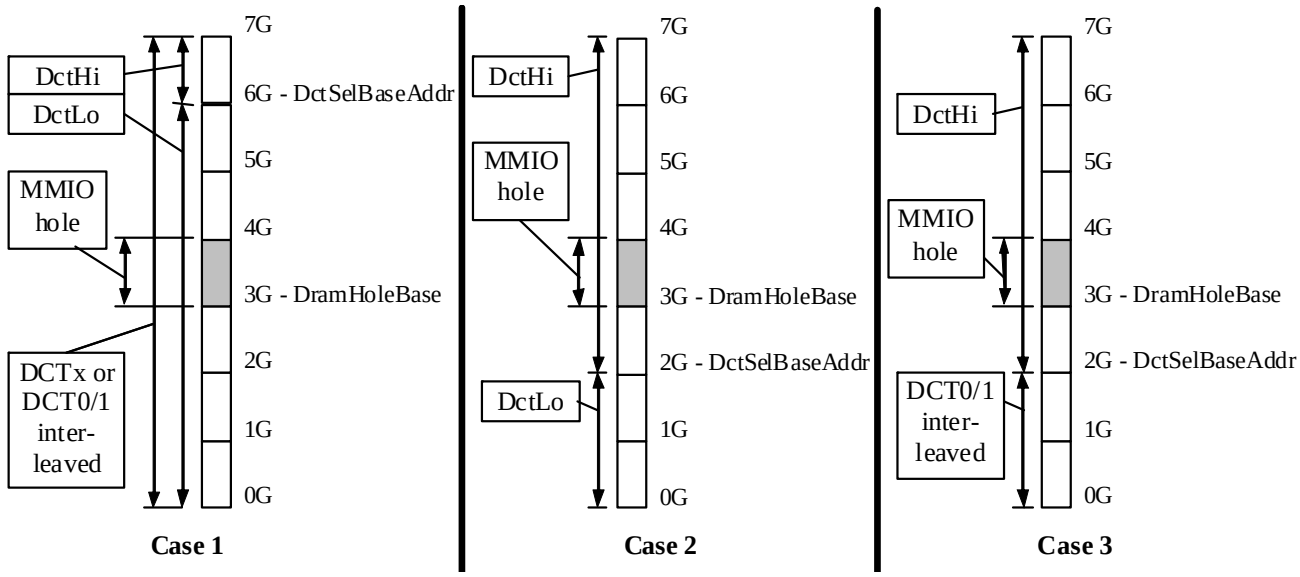


**Figure 8: Example cases for programming DramHoleOffset**

### 2.9.5.3 DctSelBaseOffset Programming

D18F2x114[DctSelBaseOffset] is programmed to one of the following equations based on the scenario:

- **Case 1**: for any of the following conditions:
  - 2 DCTs are enabled in non-interleaved mode

  then:
  - DctSelBaseOffset[39:26] = {DctSelBaseAddr[39:27], 0b};
- **Case 2**: for any of the following conditions:
  - 2 DCTs are enabled in channel interleaved mode and (DctSelBaseAddr < DramHoleBase)
  - 2 DCTs are enabled in channel interleaved mode and there is no memory hole in the address map

  then:
  - DctSelBaseOffset[39:26] = {0b,DctSelBaseAddr[39:27]};
- **Case 3**: for any of the following conditions:
  - 2 DCTs are enabled in channel interleaved mode, DctSelBaseAddr > DramHoleBase, and the interleaved range includes the MMIO hole

  then:
  - DramHoleBase[26:24] must be 000b.
  - DctSelBaseOffset[39:26] = {00h,DramHoleSize[31:26]}+{0b,(DctSelBaseAddr[39:27]–{00h,Dram-HoleSize[31:27]})};
- **Case 4**: for any of the following conditions:
  - 1 DCT is enabled (D18F2x110[DctSelHiRngEn]=0)
  - 2 DCTs are enabled in channel interleaved mode and the same size (D18F2x110[DctSelHiRngEn]=0)

  then:
  - DctSelBaseOffset[39:26] is not used.

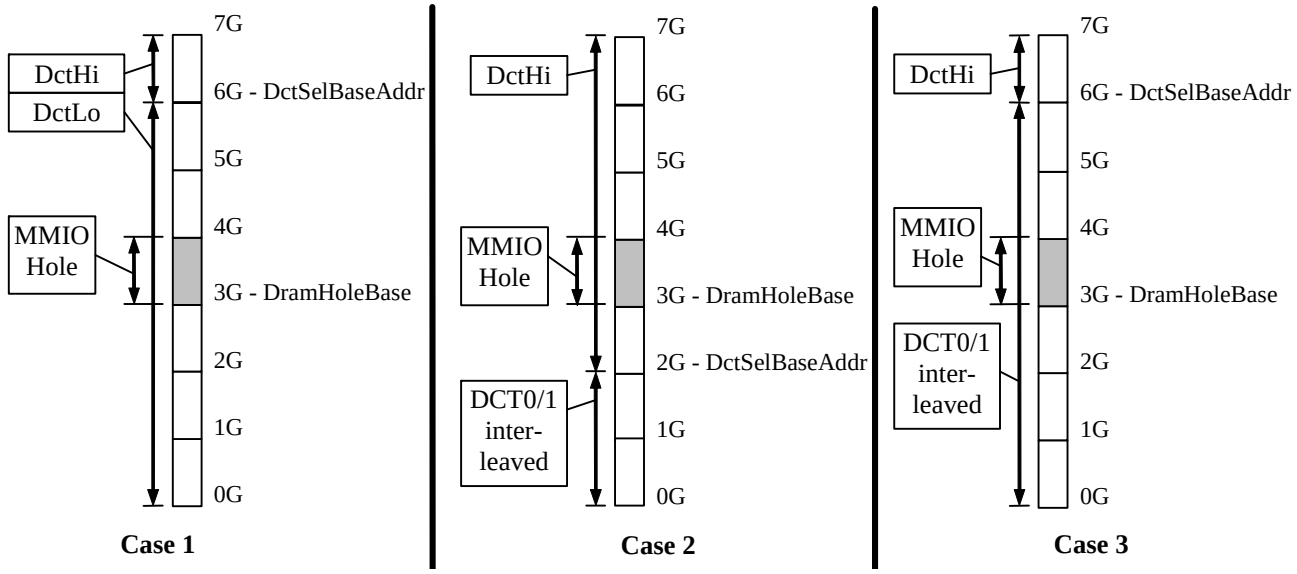


**Figure 9: Example cases for programming DctSelBaseOffset**

### 2.9.6 DRAM CC6/PC6 Storage

DRAM is used to hold the state information of cores entering the C6 power management state. As part of the system setup if CC6/PC6 is enabled, BIOS configures a special region of DRAM to hold the state information. In operation, hardware protects this region from general system accesses while allowing the cores access during C-state transitions. See 2.5.3.2.3.2 [Core C6 (CC6) State] and 2.5.3.2.3.4 [Package C6 (PC6) State].

The special DRAM storage region is defined to be a fixed 16 MB beginning at the DRAM base address specified by D18F4x12C[C6Base]. BIOS must configure the storage region at the top of the DRAM range, and adjust D18F1x44[DramLimit] downward accordingly. See Table 32.

After finalizing the system DRAM configuration, BIOS must set D18F2x118[C6DramLock] = 1.

**Table 32. C6Base Programming Example**

| DRAM Populated | D18F1x40[DramBase], D18F1x44[DramLimit] | D18F4x12C[C6Base] |
|---|---|---|
| 256 MB | 0 MB,<br>240 MB - 1 | 240 MB |

### 2.9.7 DRAM On DIMM Thermal Management

The DCTs can throttle commands and adjust the Tref refresh rate based on the state of the processor's EVENT_L pin. The recommended BIOS configuration is as follows:

- BIOS may enable DCT command throttling by programming [The DRAM Controller Temperature Throttle Register] D18F2xA4 if the platform supports the EVENT_L pin.
  - The recommended usage is for this pin to be connected to one or more JEDEC defined on DIMM temperature sensors. The DIMM SPD indicates on DIMM temperature sensor support.
  - BIOS configures the temperature sensor(s) to assert the EVENT_L pin active low when the trip point is exceeded and deassert EVENT_L when the temperature drops below the trip point minus the sensor

defined hysteresis.
  - BIOS programs D18F2xA4[ThrottleEn] with the throttling mode to employ when the trip point has been exceeded.
- BIOS configures D18F2x[1,0]8C[Tref] based on JEDEC defined temperature range options, as indicated by the DIMM SPD ROM. The two defined temperature ranges are normal (with a case temperature of 85 °C), and extended (with a case temperature of 95 °C).
  - On a channel basis, if all DIMMs support the normal temperature range, or if normal and extended temperature range DIMMs are mixed, BIOS programs D18F2x[1,0]8C[Tref] to 7.8 us and D18F2xA4[DoubleTrefRateEn] =1. BIOS configures the temperature sensor trip point for all DIMMs according to the 85 °C case temperature specification.
  - On a channel basis, if all DIMMs support the extended temperature range, BIOS has two options:
    a. Follow the recommendation for normal temperature range DIMMs.
    b. Program D18F2x[1,0]8C[Tref] to 3.9 us and configure the temperature sensor trip point for all DIMMs according to the 95 °C case temperature specification.
- At startup, BIOS determines if the DRAMs are hot before enabling a DCT and delays for an amount of time to allow the devices to cool under the influence of the thermal solution. This is accomplished by checking the temperature status in the temperature sensor of each DIMM.
- The latched status of the EVENT_L pin can be read by system software in D18F2xAC[MemTempHot].

The relationship between DRAM case temperatures, trip points, and EVENT_L pin sampling intervals is outlined as follows:
- The trip points for each DIMM are configured to the case temperature specification minus a guardband temperature for the DIMM, except in the case of mixed extended temperature DIMM types noted above.
- The temperature guardband is vendor defined and is used to account for sensor inaccuracy and platform thermal design.

## 2.10 Thermal Functions

Thermal functions SB-TSI, HTC, PROCHOT_L and THERMTRIP are intended to maintain processor temperature in a valid range by:

- Providing a signal to external circuitry for system thermal management like fan control.
- Lowering power consumption by switching to lower-performance P-state.
- Sending processor to the THERMTRIP state to prevent physical damage.

The processor thermal-related circuitry includes (1) the temperature calculation circuit (TCC) for determining the temperature of the processor and (2) logic that uses the temperature from the TCC.

### 2.10.1 The Tctl Temperature Scale

Tctl is a processor temperature control value used for processor thermal management. Tctl is accessible through SB-TSI and D18F3xA4[CurTmp]. Tctl is a temperature on its own scale aligned to the processors cooling requirements. Therefore Tctl does not represent a temperature which could be measured on the die or the case of the processor. Instead, it specifies the processor temperature relative to the maximum operating temperature, Tctl,max. Tctl,max is specified in the power and thermal data sheet. Tctl is defined as follows for all parts:

A: For Tctl = 0 to Tctl_max - 0.125: the temperature of the part is [Tctl_max - Tctl] under the maximum operating temperature.

B: For Tctl = Tctl_max to 255.875: the temperature of the part is [Tctl - Tctl_max] over the maximum operating

temperature. The processor may take corrective actions that affects performance, such as HTC, to support the return to Tctl range A.

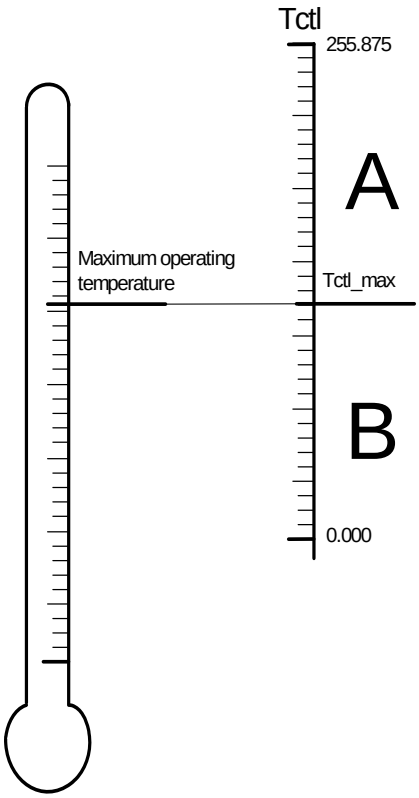


**Figure 10: Tctl scale**

### 2.10.2 Sideband Temperature Sensor Interface (SB-TSI)

The SB-TSI is used by an external SMBus master to access the internal temperature sensor and to specify temperature thresholds. The processor has access to the SB-TSI registers via [The SBI Address Register] D18F3x1E8 and [The SBI Data Register] D18F3x1EC. See SBI Temperature Sensor Interface (SB-TSI) Specification, #40821.100 kHz standard-mode and 400 kHz fast-mode are supported. 3.4 MHz high-speed mode is not supported.

### 2.10.3 Temperature-Driven Logic

The temperature calculated by the TCC is used by HTC, THERMTRIP, PROCHOT_L, and the serial interface, SB-TSI.

#### 2.10.3.1 PROCHOT_L and Hardware Thermal Control (HTC)

The processor *HTC-active state* is characterized by (1) the assertion of PROCHOT_L, (2) reduced power consumption, and (3) reduced performance. While in the HTC-active state, the processor reduces power consumption by limiting all cores to a P-state specified by D18F3x64[HtcPstateLimit]. See 2.5.3.1 [Core P-states]. While in the HTC-active state, software should not change D18F3x64 except for HtcActSts or HtcEn. Any change to fields in D18F3x64 other than HtcActSts or HtcEn while in the HTC-active state can result in undefined behavior. HTC status and control is provided through D18F3x64.

The PROCHOT_L pin acts as both an input and as an open-drain output. As an output, PROCHOT_L is driven

low to indicate that the HTC-active state has been entered due to an internal condition, as described by the following text. The minimum assertion and deassertion time for PROCHOT_L is 200 us.

The processor enters the HTC-active state if all of the following conditions are true:
- D18F3xE8[HtcCapable]=1.
- D18F3x64[HtcEn]=1.
- PWROK is asserted.
- THERMTRIP_L is deasserted.
- The processor is not in the package C1 (PC1) state or package C6 (PC6) state.

and any of the following conditions are true:
- Tctl is greater than or equal to the HTC temperature limit (D18F3x64[HtcTmpLmt]).
- PROCHOT_L is asserted.

The processor exits the HTC-active state when all of the following are true:
- Tctl is less than the HTC temperature limit (D18F3x64[HtcTmpLmt]).
- Tctl has become less than the HTC temperature limit (D18F3x64[HtcTmpLmt]) minus the HTC hysteresis limit (D18F3x64[HtcHystLmt]) since being greater than or equal to the HTC temperature limit (D18F3x64[HtcTmpLmt]).
- PROCHOT_L is deasserted.

If D18F3x64[HtcEn] is cleared from 1 to 0 while the processor is in the HTC-active state and PROCHOT_L is asserted, the processor remains in the HTC-active state until PROCHOT_L is deasserted. Clearing D18F3x64[HtcEn] to 0 while PROCHOT_L is deasserted causes the processor to exit the HTC-active regardless of the state of the other exit criteria.

The default value of the HTC temperature threshold (Tctl_max) is specified in the Power and Thermal Datasheet.

#### 2.10.3.2 Local Hardware Thermal Control (LHTC)

The *LHTC-active state* is characterized by (1) reduced power consumption and (2) reduced performance. While in the LHTC-active state, the processor reduces power consumption by limiting the maximum P-state specified by D18F3x138[LHtcPstateLimit]. See 2.5.3.1 [Core P-states]. LHTC status and control is provided through D18F3x138. While in the LHTC-active state, software should not change D18F3x138 (except for LHtcActSts and LHtcEn). Any change to the previous list of fields when in the LHTC-active state can result in undefined behavior.

The LHTC trip point is specified by D18F3x138[LHtcTmpLmt]. The LHTC-active state is independent from the HTC-active state. LHTC does not affect the PROCHOT_L output and is not affected by the PROCHOT_L input.

The processor enters the LHTC-active state if all of the following conditions are true:
- D18F3xE8[LHtcCapable]=1
- D18F3x138[LHtcEn]=1
- PWROK is asserted.
- The processor is not in the package C1 (PC1) state or package C6 (PC6) state.
- Tctl is greater than or equal to the LHTC temperature limit (D18F3x138[LHtcTmpLmt]).

A core exits the LHTC-active state when the following is true:
- Tctl is less than the LHTC temperature limit (D18F3x138[LHtcTmpLmt]) minus the LHTC hysteresis (D18F3x138[LHtcHystLmt]) since being greater than or equal to the LHTC temperature limit

(D18F3x138[LHtcTmpLmt]).

#### 2.10.3.3 THERMTRIP

If the processor supports the THERMTRIP state (as specified by [The Thermtrip Status Register] D18F3xE4[ThermtpEn] or CPUID Fn8000_0007_EDX[TTP], which are the same) and the temperature approaches the point at which the processor may be damaged, the processor enters the THERMTRIP state. The THERMTRIP state is characterized as follows:

- The THERMTRIP_L signal is asserted.
- The RESET_L signal is not asserted.
- Nearly all clocks are gated off to reduce dynamic power.
- A low-value VID is generated.
- In addition, the external chipset is expected to place the system into the S5 ACPI state (power off) if THERMTRIP_L is detected to be asserted.

A cold reset is required to exit the THERMTRIP state.

## 2.11 Root complex

### 2.11.1 Overview

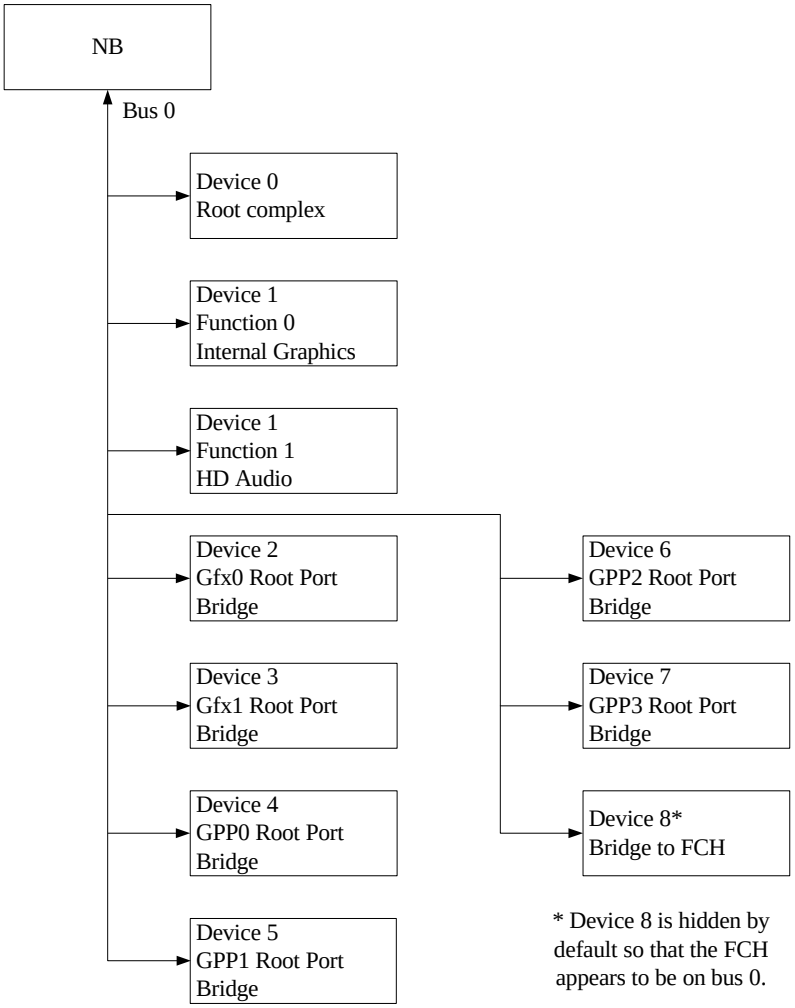


**Figure 11: Root complex topology**

### 2.11.2 Interrupt Routing

The RC remaps PCI defined INTx interrupts based on the device number of the virtual bridge or internal device that the interrupt is received from.

**Table 33: INTx Mapping**

| Device number | INTA | INTB | INTC | INTD |
|---|---|---|---|---|
| 1 | INTC | INTD | INTA | INTB |
| 2 | INTC | INTD | INTA | INTB |
| 3 | INTD | INTA | INTB | INTC |
| 4 | INTA | INTB | INTC | INTD |
| 5 | INTB | INTC | INTD | INTA |
| 6 | INTC | INTD | INTA | INTB |
| 7 | INTD | INTA | INTB | INTC |

### 2.11.3 Links

#### 2.11.3.1 Overview

There are 7 configurable ports, which can be divided into 2 groups:
- Gfx: Contains 2 x8 ports. Each port can be limited to lower link widths for applications that require fewer lanes. Additionally, the two ports can be combined to create a single x16 link.
- GPP: Contains 1 x4 UMI and 4 General Purpose Ports (GPP).

Gfx and GPP ports each have a Type 1 Virtual PCI-to-PCI bridge header in the PCI configuration space mapped to devices according to Figure 11.

Each PCIe and DDI lane is assigned a unique lane ID that software uses to communicate configuration information to the SMU. Table 34 shows the mappings between lane ID's and lanes.

**Table 34: Lane Id Mapping**

| Lane Id | Lane | Lane Id | Lane | Lane Id | Lane | Lane Id | Lane |
|---|---|---|---|---|---|---|---|
| 0 | P_UMI_[T,R]X[P,N]0 | 8 | P_GFX_[T,R]X[P,N]0 | 16 | P_GFX_[T,R]X[P,N]8 | 24 | DP1_TX[P,N]0 |
| 1 | P_UMI_[T,R]X[P,N]1 | 9 | P_GFX_[T,R]X[P,N]1 | 17 | P_GFX_[T,R]X[P,N]9 | 25 | DP1_TX[P,N]1 |
| 2 | P_UMI_[T,R]X[P,N]2 | 10 | P_GFX_[T,R]X[P,N]2 | 18 | P_GFX_[T,R]X[P,N]10 | 26 | DP1_TX[P,N]2 |
| 3 | P_UMI_[T,R]X[P,N]3 | 11 | P_GFX_[T,R]X[P,N]3 | 19 | P_GFX_[T,R]X[P,N]11 | 27 | DP1_TX[P,N]3 |
| 4 | P_GPP_[T,R]X[P,N]0 | 12 | P_GFX_[T,R]X[P,N]4 | 20 | P_GFX_[T,R]X[P,N]12 | 28 | DP0_TX[P,N]0 |
| 5 | P_GPP_[T,R]X[P,N]1 | 13 | P_GFX_[T,R]X[P,N]5 | 21 | P_GFX_[T,R]X[P,N]13 | 29 | DP0_TX[P,N]1 |
| 6 | P_GPP_[T,R]X[P,N]2 | 14 | P_GFX_[T,R]X[P,N]6 | 22 | P_GFX_[T,R]X[P,N]14 | 30 | DP0_TX[P,N]2 |
| 7 | P_GPP_[T,R]X[P,N]3 | 15 | P_GFX_[T,R]X[P,N]7 | 23 | P_GFX_[T,R]X[P,N]15 | 31 | DP0_TX[P,N]3 |

#### 2.11.3.2 Link Configurations

Lanes of the Gfx ports can be assigned to IO links or DDI links.
The following link configurations are supported for the Gfx links:

**Table 35: Supported Gfx Port Configurations**

| D0F0xE4 | | | | | | | Gfx Port Lanes | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| x0131_0080 | x0111_0011 | x0211_0011 | x0131_8021 | x0131_8022 | x0131_8040 | x0131_8013 | 3:0 | 7:4 | 11:8 | 15:12 |
| 0000_0000h | 0200_0000h | 0200_0000h | 7654_3210h | 7654_3210h | 0000_0000h | 000x_x001h | x16 Link | | | |
| 0000_0005h | 0001_0000h | 0001_0000h | 7654_3210h | 7654_3210h | 0000_0000h | 000x_x001h | x8 Link | | x8 Link | |
| 0000_0005h | 0001_0000h | 0000_0300h | xxxx_3210h | xxxx_3210h | 0000_xx0Ch | 000x_x001h | x8 Link | | DDI | DDI |

**Table 35: Supported Gfx Port Configurations**

| D0F0xE4 | | | | | | | Gfx Port Lanes | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| x0131_0080 | x0111_0011 | x0211_0011 | x0131_8021 | x0131_8022 | x0131_8040 | x0131_8013 | 3:0 | 7:4 | 11:8 | 15:12 |
| 0000_0005h | 0001_0000h | 0001_0000h | xxxx_3210h | xxxx_3210h | 0000_xx0Ch | 000x_x001h | x8 Link | | Dual-DVI | |
| 0000_0005h | 0001_0000h | 0001_0000h | xxxx_7654h | 3210_xxxxh | 0000_xx03h | 000x_x004h | Dual-DVI | | x8 Link | |
| 0000_0005h | 0000_0300h | 0001_0000h | xxxx_7654h | 3210_xxxxh | 0000_xx03h | 000x_x004h | DDI | DDI | x8 Link | |
| 0000_0005h | 0000_0300h | 0000_0300h | xxxx_3210h | xxxx_3210h | 0000_xx0Ch | 000x_x001h | x4 Link | x4 Link | DDI | DDI |
| 0000_0005h | 0000_0300h | 0001_0000h | xxxx_3210h | xxxx_3210h | 0000_xx0Ch | 000x_x001h | x4 Link | x4 Link | Dual-DVI | |
| 0000_0005h | 0000_0300h | 0000_0300h | xxxx_xx76h | 10xx_xxxxh | 0000_xx07h | 000x_x008h | DDI | DDI | DDI | x4 Link |
| 0000_0005h | 0001_0000h | 0000_0300h | xxxx_xx76h | 10xx_xxxxh | 0000_xx07h | 000x_x008h | Dual-DVI | | DDI | x4 Link |
| xxxx_xxxxh | 0000_0300h | 0000_0300h | xxxx_xxxxh | xxxx_xxxxh | 0000_xx0Fh | 000x_x000h | DDI | DDI | DDI | DDI |
| xxxx_xxxxh | 0000_0300h | 0001_0000h | xxxx_xxxxh | xxxx_xxxxh | 0000_xx0Fh | 000x_x000h | DDI | DDI | Dual-DVI | |
| xxxx_xxxxh | 0001_0000h | 0000_0300h | xxxx_xxxxh | xxxx_xxxxh | 0000_xx0Fh | 000x_x000h | Dual-DVI | | DDI | DDI |
| xxxx_xxxxh | 0001_0000h | 0001_0000h | xxxx_xxxxh | xxxx_xxxxh | 0000_xx0Fh | 000x_x000h | Dual-DVI | | Dual-DVI | |
| 0000_0005h | 0000_0300h | 0000_0300h | xxxx_xx10h | xxxx_xx10h | 0000_xx0Eh | 000x_x001h | x4 Link | DDI | DDI | DDI |
| 0000_0005h | 0000_0300h | 0001_0000h | xxxx_xx10h | xxxx_xx10h | 0000_xx0Eh | 000x_x001h | x4 Link | DDI | Dual-DVI | |
| 0000_0005h | 0001_0000h | 0000_0300h | xx76_3210 | 54xx_3210 | 0000_xx04h | 000x_x001h | x8 Link | | DDI | x4 Link |
| 0000_0005h | 0000_0300h | 0001_0000h | xx10_7654h | 3210_xx54h | 0000_xx02h | 000x_x001h | x4 Link | DDI | x8 Link | |

The following link configurations are supported for the GPP links:

**Table 36: Supported General Purpose (GPP) Link Configurations**

| D0F0xE4 | | | GPP Port Lane | | | |
|---|---|---|---|---|---|---|
| x0130_0080 | x0110_0011 | D0F0x64_x20[IocPcieDevRemapDis] | 0 | 1 | 2 | 3 |
| 0000_0001h | 0000_0300h | x | x4 Link (Device 4) | | | |
| 0000_0002h | 0000_010Ch | 0 | x2 Link (Device 4) | | x2 Link (Device 5) | |
| 0000_0003h | 0000_0104h | 0 | x2 Link (Device 4) | | x1 Link (Device 5) | x1 Link (Device 6) |
| 0000_0004h | 0000_0100h | x | x1 Link (Device 4) | x1 Link (Device 5) | x1 Link (Device 6) | x1 Link (Device 7) |
| 0000_0002h | 0000_010Ch | 1 | x2 Link (Device 4) | | x2 Link (Device 6) | |
| 0000_0003h | 0000_0104h | 1 | x2 Link (Device 4) | | x1 Link (Device 6) | x1 Link (Device 7) |

#### 2.11.3.3 Clocking

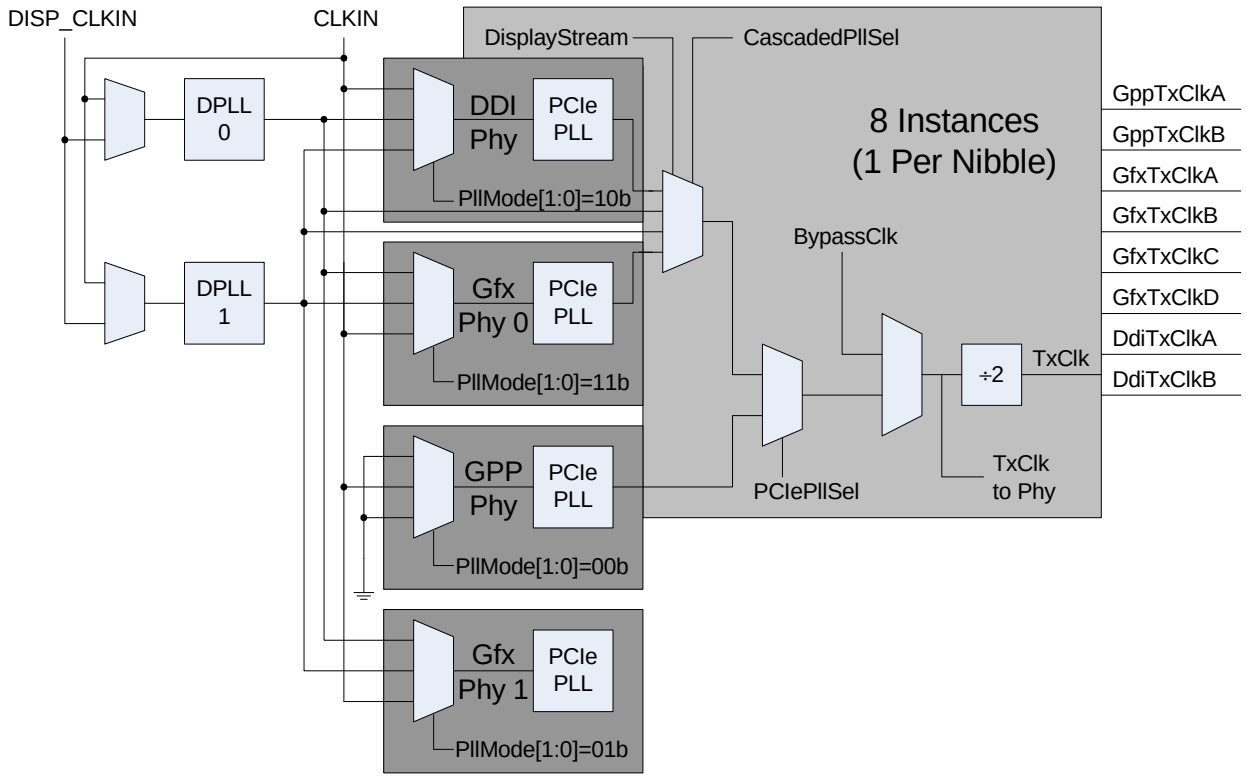


**Figure 12: Phy clock configuration**

Each phy contains a PLL and 2 clock picker circuits. Each PLL can be controlled independently of the phy and has its clock signal routed to the phys. The PLL in Gfx[8:15] is not connected and should be turned off to conserve power. The GPP PLL is always placed in PCI Express mode and is used as the transmit clock (TxClk) for all PCI Express links. The remaining two PLLs can be configured as clocks for DDI modes.

Setting up the link clocking scheme includes:
- Configuring PLL behavior. See 2.11.4.2.1 [Clock Configuration].
- Selecting the clock source for phys, DDI streams, cores, and phy interface.

### 2.11.4 Root Complex Configuration

#### 2.11.4.1 LPC MMIO Requirements

To ensure proper operation of LPC generated DMA requests, the UMI must be configured to send processor generated MMIO writes that target the LPC bus to the FCH as non-posted writes. To ensure this requirement the MMIO address space of the LPC bus must not be included in the ranges specified by [The Memory Mapped IO Base Registers] D18F1x[B8,B0,A8,A0,98,90,88,80] and [The Memory Mapped IO Limit Registers] D18F1x[BC,B4,AC,A4,9C,94,8C,84] and non-posted protocol for memory writes must be enabled using the following sequence before LPC DMA transactions are initiated.

1. Configure the FCH to use the non-posted write protocol. See the FCH register specification for configuration details.
2. Program D0F0x98_x06[UmiNpMemWrEn] = 1.

### 2.11.4.2 Link Configuration and Initialization

The overall link configuration and initialization process can be summarized into the following parts:
- Clock Configuration (see 2.11.3.3 [Clocking] and 2.11.4.2.1 [Clock Configuration]).
- 2.11.4.2.2 [Link Configuration and Core Initialization]
- 2.11.4.2.3 [Link Training]
- 2.11.4.4 [Power Management]
- Lock link configuration registers.
  - Program D0F0xE4_x0[2:1]01_0010[HwInitWrLock] = 1.
  - Program D0F0x64_x00[HwInitWrLock] = 1.
- Hide UMI link configuration space
  - Program D0F0x64_x00[NbFchCfgEn] = 1.

#### 2.11.4.2.1 Clock Configuration

Clock configuration is required after each cold or warm reset performed. The configuration applies to all phys, regardless of being configured as an IO link or DDI link.

1. Program the PLL to be powered off:
   A. Program D0F0xE4_x0[2:1]1[2:0]_001[3:2][PllPowerStateInOff]=111b.
   B. Program D0F0xE4_x0[2:1]1[2:0]_001[3:2][PllRampUpTime]=000b.
   C. Program D0F0xE4_x0[2:1]1[2:0]_0010[Ls2ExitTime]=000b.
2. Power down sub-link PLLs:
   A. Program D0F0xE4_x0130_8023[LaneEnable]=0Fh. See D0F0xE4_x013[2:0]_8023
   B. Program D0F0xE4_x0131_8023[LaneEnable]=00h.
   C. Program D0F0xE4_x0132_8023[LaneEnable]=00h.
3. Wait for D0F0xE4_x0[2:1]1[2:0]_0015[7:0]==FFh.
4. Program the PLL mode:
   A. Program D0F0xE4_x0121_2005[PllMode]=11b. See D0F0xE4_x0[2:1]2[2:0]_2005
   B. Program D0F0xE4_x0221_2005[PllMode] = 01b.
   C. Program D0F0xE4_x0122_2005[PllMode] = 10b.
5. Enable the individual lanes:
   A. Program D0F0xE4_x0130_8023[LaneEnable]=FFh. See D0F0xE4_x013[2:0]_8023
   B. Program D0F0xE4_x0131_8023[LaneEnable]=FFFFh.
   C. Program D0F0xE4_x0132_8023[LaneEnable]=FFh.
   D. Wait for D0F0xE4_x0[2:1]1[2:0]_0015[7:0]==FFh.
6. Program D0F0xE4_x0[2:1]1[2:0]_001[3:2][PllPowerStateInOff]=000b.

#### 2.11.4.2.2 Link Configuration and Core Initialization

Link configuration is done on a per link basis. Lane reversal, IO link/DDI link selection, and lane enablement is configured through this sequence.

1. Place software-reset module into blocking mode:
   A. Program D0F0xE4_x013[2:0]_8062[ConfigXferMode]=0.
   B. Program D0F0xE4_x013[2:0]_8062[BlockOnIdle]=0.
2. If the link is an IO link, Program D0F0xE4_x0[2:1]01_0011[DynClkLatency]=Fh.
3. Program D0F0xE4_x013[1:0]_0080 per Table 35 and Table 36.
4. Program D0F0xE4_x013[2:0]_8021 per Table 35 and Table 36.
5. Program D0F0xE4_x0[2:1]1[2:0]_0010[RxDetectTxPwrMode]=1.
6. Program D0F0xE4_x0[2:1]1[2:0]_0010[Ls2ExitTime]=000b.

7. Program D0F0xE4_x013[2:0]_8013[MasterPciePllA, MasterPciePllB, MasterPciePllC, MasterPciePllD] per Table 35.
8. Initiate core reconfiguration sequence:
   A. Program D0F0xE4_x013[2:0]_8062[ReconfigureEn]=1.
   B. Program D0F0xE4_x013[2:0]_8060[Reconfigure]=1.
   C. Wait for D0F0xE4_x013[2:0]_8060[Reconfigure]==0.
   D. Program D0F0xE4_x013[2:0]_8062[ReconfigureEn]=0.
9. Return software-reset module to non-blocking mode:
   A. Program D0F0xE4_x013[2:0]_8062[ConfigXferMode]=1.
10. Program D[8:2]F0xE4_xC1[StrapReverseLanes] if necessary.
11. Program D0F0xE4_x0[2:1]1[2:0]_0011 per Table 35 and Table 36.
12. Program D0F0xE4_x0[2:1]2[2:0]_[7:6][7:6,3:0][8,0]5 per Table 64.
13. For each link mapped to DDI:
   A. Program D0F0xE4_x0[2:1]1[2:0]_001[3:2][PllPowerStateInTxs2]=111b.
   B. Program D0F0xE4_x0[2:1]1[2:0]_001[3:2][PllPowerStateInOff]=111b.
   C. Program D0F0xE4_x0[2:1]1[2:0]_001[3:2][PllRampUpTime]=010b.
14. For each nibble that has no PCIe lanes in use:
   A. Program D0F0xE4_x0[2:1]1[2:0]_001[3:2][PllPowerStateInOff]=111b.
   B. Program D0F0xE4_x0[2:1]1[2:0]_001[3:2][PllPowerStateInTxs2]=111b.
   C. Program D0F0xE4_x0[2:1]1[2:0]_001[3:2][TxPowerStateInTxs2]=111b.
   D. Program D0F0xE4_x0[2:1]1[2:0]_001[3:2][RxPowerStateInRxs2]=111b.
15. For each lane that is not in use, program the corresponding D0F0xE4_x013[2:0]_8023[LaneEnable]=0.
16. If the link is a DDI link:
   A. Program D0F0xE4_x013[2:1]_8040[OwnPhyA, OwnPhyB, OwnPhyC, OwnPhyD] per Table 35.
17. Configure PIF parings and disable ganged mode for UMI:
   A. Program D0F0xE4_x0110_0011=0000_0300h.
   B. Program D0F0xE4_x0120_6[3:2][8,0]5[GangedModeEn]=0.

#### 2.11.4.2.3 Link Training

Link training is performed on a per link basis. BIOS may train the links in parallel.

```
foreach(link)
{
   reset_count = 0;
   train_state = detect;

   while(train_state != link_done){
      D0F0xE4_x013[1:0]_0[C:8]00[HoldTraining] = 0;
      wait(2 ms);
      start timer;
      train_state = detect;

      while(train_state == detect) {
         if(D[8:2]F0xE4_xA5[LcCurrentState, LcPrevState1, LcPrevState2, LcPrevState3]==3Fh)
            train_state=assert_reset;

         if(LcCurrentState >= 04h)
            train_state=device_detected;

         if(timer >= 40 ms) //No device present
            train_state=clean_up;
      }
      if (train_state == device_detected) train_state = check_broken_lane (link);
      if (train_state == compliance) train_state = check_compliance (link);
```

```
      if (train_state == check_vc_negotiation) train_state = check_vc (link);
      if (train_state == reset_assert) train_state = reset_link (link);
      if (train_state == clean_up) train_state = power_down_link (link);
   }
}
check_broken_lane (link)
{
   found062A = 0;
   found092A = 0;
   return_code = device_detected;
   Switch(D[8:2]F0xE4_xA5[LcCurrentState, LcPrevState1, LcPrevState2, LcPrevState3])
   {
      case xxxx_062Ah:
      case xx06_xx2Ah:
      case 06xx_xx2Ah:
      case xx06_2Axxh:
      case 06xx_2Axxh:
      case 062A_xxxxh: found062A = 1;
                       break;
      case xxxx_092Ah:
      case xx09_xx2Ah:
      case 09xx_xx2Ah:
      case xx09_2Axxh:
      case 09xx_2Axxh:
      case 092A_xxxxh: found092A = 1;
                       break;
              default: found062A = 0;
                       found092A = 0;

   }
   if(found062A == 1 && D[8:2]F0xE4_xA2[LcLinkWidthRd] != expected_width) {
      disable_broken_lane (link);
      expected_width = D[8:2]F0xE4_xA2[LcLinkWidthRd];
      return_code = assert_reset;
   }
   else if (found062A == 1 || found092A == 1){
      D[8:2]F0x88[TargetLinkSpeed] = 1h;
      D[8:2]F0x88[HwAutonomousSpeedDisable] = 1;
      D[8:2]F0xE4_xA4[LcGen2EnStrap] = 0;
      D[8:2]F0xE4_xA4[LcMultUpstreamAutoSpdChngEn] = 0;
      D[8:2]F0xE4_xC0[StrapAutoRcSpeedNegotiationDis] = 1;
      D[8:2]F0xE4_xA2[LcUpConfigureDis] = 1;
      D0F0xE4_x013[1:0]_0[C:8]03[StrapBifDeemphasisSel] = 0;
      return_code = assert_reset;
   }
   else {
      return_code = compliance;
   }
   return (return_code);
}

check_compliance (link)
{
   start timer // do not poll for more than 2 s
   return_code = compliance;
   while(return_code == compliance){
      if(D[8:2]F0xE4_xA5[LcCurrentState] == 07h) return_code = link_done;
      if(D[8:2]F0xE4_xA5[LcCurrentState] == 10h ) return_code = vc_negotiation;
      if (timer >= 2s) return_code = assert_reset;
   }
```

```
   return (return_code);
}

check_vc (link, Reset)
{
   return_code = link_done;
   if(D[8:2]F0x128[VcNegotiationPending] == 1){
      D[8:2]F0xE4_xA2[LcShortReconfigEn] = 0;
      D[8:2]F0xE4_xA2[LcReconfigNow] = 1;
      wait (5ms)
      return_code = detect;
   }
   return (return_code);
}

reset_link(link, reset_count)
{
   if (reset_count < 15) {
      reset_count ++;
      D0F0x64_x0C[DevXBridgeDis] = 1;
      D[8:2]F0xE4_xA0[LcResetLink] = 1;
      wait (1ms)
      D[8:2]F0xE4_xA0[LcResetLink] = 0;
      return_code = detect;
   }
   else {
      return_code = clean_up;
   }
   return (return_code);
}

clean_up(link)
{
   if (link does not support hot plug) {
      D0F0xE4_x013[1:0]_0[C:8]00[HoldTraining] = 1;
      D0F0x64_x0C[DevXBridgeDis] = 1;
   }
   return (link_done);
}

disable_broken_lane (link)
{
   D0F0xE4_x013[2:0]_8023[LaneEnable] = 0;
   D0F0xE4_x0[2:1]1[2:0]_001[3:2][PllPowerStateInOff] = 07h;
   D0F0xE4_x0[2:1]1[2:0]_001[3:2][PllPowerStateInTxs2] = 07h;
   D0F0xE4_x0[2:1]1[2:0]_001[3:2][TxPowerStateInTxs2] = 07h;
   D0F0xE4_x0[2:1]1[2:0]_001[3:2][RxPowerStateInRxs2] = 07h;
}
```

#### 2.11.4.3 Miscellaneous Features

##### 2.11.4.3.1 Straps

1. Program D0F0xE4_x013[2:0]_8011[StrapBifValid]=1.
2. Program strap values.
3. Program D0F0xE4_x013[2:0]_8011[StrapBifValid]=0.

#### 2.11.4.3.2 Lane Reversal

Normally, the lanes of each port are physically numbered from n-1 to 0 where n is the number of lanes assigned to the port. Physical lane numbering can be reversed according to the following methods:

- To reverse the physical lane numbering for a specific port, program D[8:2]F0xE4_xC1[StrapReverse-Lanes]=1 according to the sequence in 2.11.4.3.1 [Straps].
- To reverse the physical lane numbering for all ports in the GPP or GFX interfaces, program D0F0xE4_x0[2:1]01_00C0[StrapReverseAll]=1 according to the sequence in 2.11.4.3.1 [Straps].

Note that logical port numbering is established during link training regardless of the physical lane numbering.

#### 2.11.4.3.3 Link Speed Changes

Link speed changes can only occur on Gen2 capable links. To verify that Gen2 speeds are supported verify D[8:2]F0x64[LinkSpeed]==02h.

##### 2.11.4.3.3.1 Software Initiated Link Speed Changes

The following programming sequence describes a software initiated speed change from Gen1 to Gen2:

1. Verify D[8:2]F0xE4_xA4[LcGen2EnStrap]==1. If LcGen2EnStrap is not 1, program it to 1.
2. Program D[8:2]F0x88[TargetLinkSpeed]=2.
3. Program D[8:2]F0xE4_xA4[LcGoToRecovery]=1.
4. Verify D[8:2]F0xE4_xA4[LcOtherSideSupportsGen2]==1, otherwise stop.
5. If D[8:2]F0x68[LinkSpeed]==2h, stop. The link is already at Gen2 speed.
6. If D[8:2]F0xE4_xA4[LcSpeedChangeAttemptFailed]==1, stop. The maximum number of speed negotiation failures have been reached.
7. Ensure D[8:2]F0xE4_xA4[LcForceDisSwSpeedChange]=0.
8. Ensure D[8:2]F0x88[HwAutonomousSpeedDisable]=0.
9. Program D[8:2]F0xE4_xA4[LcInitiateLinkSpeedChange]=1.

The following programming sequence describes a software initiated speed change from Gen2 to Gen1:

1. Program D[8:2]F0x88[TargetLinkSpeed]=1h.
2. If D[8:2]F0x68[LinkSpeed]==1h, stop. The link is already at Gen1 speed.
3. If D[8:2]F0xE4_xA4[LcSpeedChangeAttemptFailed]==1, stop. The maximum number of speed negotiation failures have been reached.
4. Ensure D[8:2]F0xE4_xA4[LcForceDisSwSpeedChange]=0.
5. Ensure D[8:2]F0x88[HwAutonomousSpeedDisable]=0.
6. Program D[8:2]F0xE4_xA4[LcInitiateLinkSpeedChange]=1.

##### 2.11.4.3.3.2 Autonomous Link Speed Changes

To enable autonomous speed changes on a per port basis:

1. Program D[8:2]F0x88[TargetLinkSpeed]=2h.
2. Program D0F0xE4_x013[1:0]_0[C:8]03[StrapBifDeemphasisSel]=1.
3. Program D[8:2]F0xE4_xA4[LcGen2EnStrap]=1.
4. Program D[8:2]F0xE4_xC0[StrapAutoRcSpeedNegotiationDis]=0.
5. Program D[8:2]F0xE4_xA4[LcMultUpstreamAutoSpdChangEn]=1.
6. Program D[8:2]F0xE4_xA2[LcUpconfigureDis]=0.

#### 2.11.4.3.4 Deemphasis

Deemphasis strength can be changed on a per-port basis by programming D[8:2]F0xE4_xB5[LcSelectDeem-

phasis].

#### 2.11.4.4 Power Management

##### 2.11.4.4.1 Link States

To enable support for L1 program D[8:2]F0xE4_xA0[LcL1Inactivity]=6h.

To enable support for L0s:
- Program D[8:2]F0xE4_xA1[LcDontGotoL0sifL1Armed]=1.
- Program D[8:2]F0xE4_xA0[LcL0sInactivity]=9h.

##### 2.11.4.4.2 Dynamic Link-width Control

Dynamic link-width control is a power saving feature that reconfigures the link to run with fewer lanes. The inactive lanes are turned off to conserve power.

The GFX links can switch among widths of: x1, x2, x4, x8, and x16.
The GPP links can switch among widths of: x1, x2, and x4.

There are 3 link-width control mechanisms:
- Long Reconfiguration: The link is brought down and retrained to a different width. This mechanism is only supported on links between AMD products.
- Short Reconfiguration: The link is retrained to a different width by going through the recovery state of the LTSSM. This mechanism is only supported on links between AMD products.
- Up/Down Reconfiguration: The link is retrained according to the PCI Express specification. This mechanism is only available between Gen2 devices.

The core has the capability to turn off the inactive lanes of trained links. To enable this feature program D[8:2]F0xE4_xA2[LcDynLanesPwrState]=11b.

The following pseudocode should be used to perform dynamic link-width control:

```
if(using up/down reconfiguration)
{
    if(D[8:2]F0xE4_xA2[LcRenegotiationSupport]==1)
    {
        D[8:2]F0xE4_xA2[LcUpconfigureSupport]=1;
        D[8:2]F0xE4_xA2[LcUpconfigureDis]=0;
    }
    else
        up/down reconfiguration not supported;
}

if(D[8:2]F0xE4_xA2[LcLinkWidth]==desired link width)
    stop; // Nothing to do
else
    D[8:2]F0xE4_xA2[LcLinkWidth]=desired link width;

if(using long reconfiguration)
    D[8:2]F0xE4_xA2[LcReconfigArcMissingEscape]=1;

if(using short reconfiguration)
{
    D[8:2]F0xE4_xA2[LcShortReconfigEn]=1;
```

```
    Refer to the device specific programming requirements.
}

if(using up/down reconfiguration)
    D[8:2]F0xE4_xA2[LcRenegotiateEn]=1;

D[8:2]F0xE4_xA2[LcReconfigNow]=1;
wait until D[8:2]F0x68[LinkTraining]==0;
wait until D[8:2]F0xE4_xA2[LcReconfigNow]==0;
wait until D[8:2]F0x128[VcNegotiationPending]==0;
 ;
if(using long reconfiguration)
    D[8:2]F0xE4_xA2[LcReconfigArcMissingEscape]=0;
```

#### 2.11.4.5 Link Test and Debug Features

##### 2.11.4.5.1 Compliance Mode

To enable Gen1 software compliance mode program D[8:2]F0xE4_xC0[StrapForceCompliance]=1 for each port to be placed in compliance mode.

To enable Gen2 software compliance mode:

1. BIOS enables Gen2 capability by programming D0F0xE4_x0[2:1]01_00C1[StrapGen2Compliance]=1.
2. Program D0F0xE4_x013[1:0]_0[C:8]03[StrapBifDeemphasisSel]=1 for each port to be placed in compliance mode.
3. Program D[8:2]F0x88[TargetLinkSpeed]=2h for each port to be placed in compliance mode.
4. Program D[8:2]F0x88[EnterCompliance]=1 for each port to be placed in compliance mode.

### 2.11.5 BIOS Timer

The root complex implements a 32-bit microsecond timer (see D0F0xE4_x0130_80F0 and D0F0xE4_x0130_80F1) that the BIOS can use to accurately time wait operations between initialization steps. To ensure that BIOS waits a minimum number of microseconds between steps BIOS should always wait for one microsecond more than the required minimum wait time.

## 2.12 System Management Unit (SMU)

The system management unit (SMU) is a subcomponent of the northbridge that is responsible for a variety of system and power management tasks during boot and runtime. Several internal registers are used to control various tasks. See 3.17 [GPU Memory Mapped Registers] for registers descriptions and details.

### 2.12.1 Microcontroller

The SMU contains a microcontroller with a 16k ROM and a 16k RAM.

#### 2.12.1.1 Updating Firmware

Software updates firmware using the following sequence:

1. Program SMUx0B_x84EC=0.
2. Interrupt the SMU with Service Index 10h. See 2.12.1.2 [Software Interrupts].
3. Wait for SMUx0B_x84EC[SaveStateDone]==1.
4. Program SMUx01[Reset, RamSwitch]=[0, 1].
5. For each dword of data:

- Program the address of the dword to SMUx0B[MemAddr].
- Program the data to SMUx05[McuRam].

6. Program SMUx01[VectorOverride, Reset, RamSwitch]=[1, 1, 0].

When the microcontroller comes out of reset it begins executing the updated firmware.

#### 2.12.1.2 Software Interrupts

BIOS or ACPI methods can interrupt the microcontroller to make firmware perform a specific task using to following sequence:

1. Wait for SMUx03[IntDone]==1.
2. If SMUx03[IntReq]==1, program SMUx03[IntReq]=0.
3. Program SMUx03[ServiceIndex] and set SMUx03[IntReq]=1. This may be done as a single write.
4. Wait for SMUx03[IntAck]==1.

After performing the steps above, software may continue execution before the interrupt has been serviced (see SMUx03[IntDone]). However, software should not rely on the results of the interrupt until the service is complete. Interrupting the SMU with a service index that does not exist results in undefined behavior.

**Table 37: Microcontroller Interrupt Service Indices**

| Service Index | Notes |
|---|---|
| 01h | Input:SMUx0B_x8408, SMUx0B_x840C, SMUx0B_x8410, SMUx0B_x84A0, SMUx0B_x84A4, SMUx0B_x84A8 |
| | Output: None. |
| | Description: Configures power gating for various GPU and RC subcomponents. See 2.5.5.4 [GPU and Root Complex Power Gating]. |
| | Firmware revision: 1.4 |
| 03h | Input: None. |
| | Output: None. |
| | Description: Used by BIOS to power UVD down. UVD power gating must be initialized before this service index is called. See 2.5.5.4.3 [UVD Power Gating]. |
| | Firmware revision: 1.4 |
| 08h | Input:SMUx0B_x8434[LclkDpmEn] |
| | Output: None. |
| | Description: If SMUx0B_x8434[LclkDpmEn]==1, calling this service index enables LCLK DPM. If SMUx0B_x8434[LclkDpmEn]==0, calling this service index disables LCLK DPM. See 2.5.5.1.3 [LCLK DPM]. |
| | Firmware revision: 1.4 |
| 0Bh | Input: See 3.16 [Fixed Configuration Space (FCR)]. |
| | Output: See 3.16 [Fixed Configuration Space (FCR)]. |
| | Description: Reads or writes a register from FCR space. |
| | Firmware revision: 1.4 |

**Table 37: Microcontroller Interrupt Service Indices**

| Service Index | Notes |
|---|---|
| 0Dh | Input: See 3.16 [Fixed Configuration Space (FCR)]. |
| | Output: See 3.16 [Fixed Configuration Space (FCR)]. |
| | Description: Reads a register from FCR space. |
| | Firmware revision: 1.4 |
| 10h | Input: None |
| | Output: None |
| | Description: Places SMU into a known state before reset. |
| | Firmware revision: 1.4 |
| 12h | Input: SMUx0B_x8580. |
| | Output: None. |
| | Description: Enables hardware to dynamically change CPB power limits to account for individual cores or the GPU being idle. See 2.5.3.1.1 [Core Performance Boost (CPB)]. |
| | Firmware revision: 1.4 |
| 13h | Input: SMUx0B_x858C. |
| | Output: None. |
| | Description: Used by BIOS and ACPI methods to mark PCIe lanes unused. All lanes between SMUx0B_x858C[UpperLaneId] and SMUx0B_x858C[LowerLaneId] (inclusive) are marked. The TX and RX sides of each lane can be marked individually as specified by SMUx0B_x858C[Tx, Rx]. After completing this service, hardware powers off the Gfx link core if it is not needed . See 2.5.5.4.4 [Gfx Link Core Power Gating]. |
| | Firmware revision: 1.4 |
| 14h | Input: SMUx0B_x858C. |
| | Output: None |
| | Description: Used by BIOS and ACPI methods to mark PCIe lanes used. All lanes between SMUx0B_x858C[UpperLaneId] and SMUx0B_x858C[LowerLaneId] (inclusive) are marked. The TX and RX sides of each lane can be marked individually as specified by SMUx0B_x858C[Tx, Rx]. After completing this service, hardware powers on the Gfx link core if it is needed . See 2.5.5.4.4 [Gfx Link Core Power Gating]. |
| | Firmware revision: 1.4 |
| 18h | Input:SMUx0B_x859C. |
| | Output: None. |
| | Description: Used by BIOS to request register access the PLL indicated by SMUx0B_x859C[PllId] be powered on. |
| | Firmware revision: 1.4 |

### 2.12.2 BIOS Requirements for SMU Initialization

1. If (!GpuEnabled), program FCRxFF30_0398[SoftResetUvd, SoftResetRlc, SoftResetMc, SoftResetGrbm, SoftResetDc] = 1.
2. Update firmware if necessary (see 2.12.1.1 [Updating Firmware]).

## 2.13 Digital Display Interface (DDI)

The processor supports a digital display interface that is capable of outputting display port, DVI and HDMI formats. The DDI interface supports generating two simultaneous independent display outputs.

The processor contains a dedicated 2 port, 8 lane display interface. Links of the Gfx port can also be programmed for use as DDI interfaces. See Table 35 for the possible Gfx port configurations.

The following DDI modes are supported:

**Table 38: Supported DDI Modes**

| Mode | Bit Rate | PLL Requirements | D0F0xE4_x0[2:1]2[2:0]_000[A:9] |
|---|---|---|---|
| DisplayPort | 2.7G | Single (Display) PLL | 0[0,2]00_0000h[1] |
| DisplayPort | 1.62G | Single (Display) PLL | 0[0,2]00_0000h |
| HDMI/DVI Single Link Coherent | 250M-2.25G | Single (Display) PLL | 0[0,2]00_0000h[1] |
| HDMI/DVI Single Link Incoherent | 250M-2.25G | 2 Cascaded PLLs | 1[0,2]00_0000h[1] |
| DVI Dual Link Coherent | 770M-1.65G | Single (Display) PLL | 0[0,2]10_0000h[1] |
| DVI Dual Link Incoherent | 770M-1.65G | 2 Cascaded PLLs | 1[0,2]10_0000h[1] |

1. Bit 25 selects between PLL0 and PLL1.

There are 2 clock modes available for HDMI/DVI:
- Incoherent mode uses cascaded PLLs to ensure that data path jitter matches the clock path jitter.
- Coherent mode uses a single PLL to provide a low jitter and a low power clock source.

## 2.14 Graphics Processor (GPU)

The processor contains an integrated DX11 compliant graphics processor.

### 2.14.1 GPU PCI Interface

BIOS must configure the PCI interface block of the GPU before the GPU can be accessed. The PCI interface block is configured by programming [The Internal Graphics PCI Control 1] D0F0x64_x1C register. Two writes to this register are required. The first write sets the configuration and must have D0F0x64_x1C[WriteDis]=0. The second write uses the same write data with D0F0x64_x1C[WriteDis]=1.

### 2.14.2 Graphics Memory Controller (GMC)

The graphics memory controller is responsible for servicing memory requests from the different blocks within the GPU and forwarding routing them to the appropriate interface. The GMC is also responsible for translating GPU virtual address to GPU physical addresses and for translating GPU physical addresses to system addresses.

#### 2.14.2.1 GMC Initialization

The GMC must be initialized by BIOS in the following sequence:

1. Disable GMC clock gating.
   - Program GMMx20[C0:B8][Enable] = 0.

- Program GMMx2478[Enable] = 0.
- Program GMMx26[58:50][Enable] = 0.
- Program GMMx15C0[Enable] = 0.

2. Configure GMC registers.
3. Initialize the register save/restore engine. See 2.14.2.1.1.1 [RENG Initialization].
4. Lock GMC registers.
   - Program GMMx2B98[CriticalRegsLock] = 1.
5. Enable requests from GMC clients.
   - Program GMMx25C0 = 0000_0000h.
   - Program GMMx20EC = 0000_01FCh.
   - Program GMMx20D4 = 0000_0016h.
6. Enable GMC clock gating and power management.
   - Program GMMx20[C0:B8][Enable] = 1.
   - Program GMMx2478[Enable] = 1.
   - Program GMMx26[58:50][Enable] = 1.
   - Program GMMx15C0[Enable] = 1.
   - Program FCRxFF30_01F4[CgMcdwCgttSclkOverride, CgMcbCgttSclkOverride] = 0.
   - Program FCRxFF30_01F5[CgVmcCgttSclkOverride] = 0.
   - Program GMMx2B94[RengExecuteOnPwrUp] = 1.
   - Program GMMx2B94[RengExecuteOnRegUpdate] = 1.

#### 2.14.2.1.1 Register Save/Restore Engine (RENG)

The register save/restore engine (RENG) is used to save and restore GMC register state when power gating the GMC (see 2.5.5.4.2 [GMC Power Gating]). The RENG uses a programmable RAM to specify the save/restore registers and to store the register data.

##### 2.14.2.1.1.1 RENG Initialization

The RENG RAM should be initialized with the following sequence of writes.

**Table 39: RENG Initialization Sequence**

| Step | Address | Write Data | Step | Address | Write Data | Step | Address | Write Data |
|---|---|---|---|---|---|---|---|---|
| 1 | GMMx2B8C | 0000_0000h | 23 | GMMx2B8C | 0000_0045h | 45 | GMMx2B8C | 0000_0140h |
| 2 | GMMx2B90 | 001e_0a07h | 24 | GMMx2B90 | 0001_056Fh | 46 | GMMx2B90 | 0000_0952h |
| 3 | GMMx2B8C | 0000_0020h | 25 | GMMx2B8C | 0000_0048h | 47 | GMMx2B8C | 0000_0142h |
| 4 | GMMx2B90 | 0005_0500h | 26 | GMMx2B90 | 0005_0572h | 48 | GMMx2B90 | 0001_0954h |
| 5 | GMMx2B8C | 0000_0027h | 27 | GMMx2B8C | 0000_004Fh | 49 | GMMx2B8C | 0000_0145h |
| 6 | GMMx2B90 | 0001_050Ch | 28 | GMMx2B90 | 0000_0800h | 50 | GMMx2B90 | 0009_095Ah |
| 7 | GMMx2B8C | 0000_002Ah | 29 | GMMx2B8C | 0000_0051h | 51 | GMMx2B8C | 0000_0150h |
| 8 | GMMx2B90 | 0001_051Ch | 30 | GMMx2B90 | 0026_0801h | 52 | GMMx2B90 | 0029_096Dh |
| 9 | GMMx2B8C | 0000_002Dh | 31 | GMMx2B8C | 0000_0079h | 53 | GMMx2B8C | 0000_017Bh |
| 10 | GMMx2B90 | 0003_0534h | 32 | GMMx2B90 | 004B_082Dh | 54 | GMMx2B90 | 000E_0997h |
| 11 | GMMx2B8C | 0000_0032h | 33 | GMMx2B8C | 0000_00C6h | 55 | GMMx2B8C | 0000_018Bh |
| 12 | GMMx2B90 | 0001_053Eh | 34 | GMMx2B90 | 0013_088Dh | 56 | GMMx2B90 | 1000_09A6h |
| 13 | GMMx2B8C | 0000_0035h | 35 | GMMx2B8C | 0000_00DBh | 57 | GMMx2B90 | 0000_0040h |
| 14 | GMMx2B90 | 0001_0546h | 36 | GMMx2B90 | 1000_08A1h | 58 | GMMx2B90 | 0000_0040h |

**Table 39: RENG Initialization Sequence**

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 15 | GMMx2B8C | 0000_0038h | 37 | GMMx2B90 | 0000_0040h | 59 | GMMx2B8C | 0000_018Fh |
| 16 | GMMx2B90 | 0002_054Eh | 38 | GMMx2B90 | 0000_0040h | 60 | GMMx2B90 | 0000_09A7h |
| 17 | GMMx2B8C | 0000_003Ch | 39 | GMMx2B8C | 0000_00DFh | 61 | GMMx2B8C | 0000_0191h |
| 18 | GMMx2B90 | 0001_0557h | 40 | GMMx2B90 | 0000_08A2h | 62 | GMMx2B90 | 002E_09D7h |
| 19 | GMMx2B8C | 0000_003Fh | 41 | GMMx2B8C | 0000_00E1h | 63 | GMMx2B8C | 0000_01C1h |
| 20 | GMMx2B90 | 0001_055Fh | 42 | GMMx2B90 | 005A_08CDh | 64 | GMMx2B90 | 0017_0A26h |
| 21 | GMMx2B8C | 0000_0042h | 43 | GMMx2B8C | 0000_013Dh | 65 | GMMx2B94 | 765D_9000h |
| 22 | GMMx2B90 | 0001_0567h | 44 | GMMx2B90 | 0001_094Dh | 66 | GMMx2B98 | 200C_F01Ch |

### 2.14.3 Frame Buffer (FB)

The frame buffer is defined as the portion of system memory dedicated for GPU use.

**Table 40: Recommended Frame Buffer Configurations**

| System Memory Size | Frame Buffer Size |
|---|---|
| Less than 1GB | 64 MB |
| >= 1GB & < 2GB | 256 MB |
| >=2GB | 512 MB |

## 2.15 Debug Support

.

## 2.16 Machine Check Architecture

The processor contains logic and registers to detect, log, and (if possible) correct errors in the data or control paths in each core and the Northbridge.

Refer to the *AMD64 Architecture Programmer's Manual* for an architectural overview and methods for determining the processor's level of MCA support. See 1.2 [Reference Documents].

### 2.16.1 Machine Check Registers

The presence of the machine check registers is indicated by CPUID Fn0000_0001_EDX[MCA] or CPUID Fn8000_0001_EDX[MCA]. The ability of hardware to generate a machine check exception upon an error is indicated by CPUID Fn0000_0001_EDX[MCE] or CPUID Fn8000_0001_EDX[MCE].

The machine check register set includes:
- Global status and control registers:
  - [The Global Machine Check Capabilities Register (MCG_CAP)] MSR0000_0179
  - [The Global Machine Check Status Register (MCG_STAT)] MSR0000_017A
  - [The Global Machine Check Exception Reporting Control Register (MCG_CTL)] MSR0000_017B
- Most of the machine check MSRs are organized as a 4-register-type by 6-register-bank matrix.
  - The four register types are:
    - **MCi_CTL**, The Machine Check Control Register: Enables error reporting via machine check exception (MCE). The MC*i*_CTL register in each bank must be enabled by the corresponding enable bit in

MCG_CTL (MSR0000_017B).
  - **MCi_STATUS**: The Machine Check Status Register: Logs information associated with errors.
  - **MCi_ADDR**: The Machine Check Address Register: Logs address information associated with errors.
  - **MCi_MISC**: The Machine Check Miscellaneous Registers: Log miscellaneous information associated with errors, as defined by each error type.
- The six error-reporting register banks supported are:
  - **MC0, DC**: MSR0000_04[03:00], data cache machine check registers.
  - **MC1, IC**: MSR0000_04[07:04], instruction cache machine check registers.
  - **MC2, BU**: MSR0000_04[0B:08], bus unit machine check registers.
  - **MC3, LS**: MSR0000_04[0F:0C], load-store machine check registers.
  - **MC4, NB**: MSR0000_04[13:10], Northbridge machine check registers. These MSRs are accessible from configuration space as well.
  - **MC5, FR**: MSR00000_04[17:14], fixed-issue reorder buffer machine check registers.

Once system software has determined that machine check registers exist via the CPUID instruction, MSR0000_0179 may be read to determine how many machine check banks are implemented and if [The Global Machine Check Exception Reporting Control Register (MCG_CTL)] MSR0000_017B is present.

Table 41 identifies the addresses associated with each MCA register.

**Table 41: MCA register cross-reference table**

| Register Bank (MC*i*) | MCA Register | | | | |
|---|---|---|---|---|---|
| | CTL | STATUS | ADDR | MISC | CTL_MASK |
| MC0 | MSR0000_0400 | MSR0000_0401 | MSR0000_0402 | MSR0000_0403 | MSRC001_0044 |
| MC1 | MSR0000_0404 | MSR0000_0405 | MSR0000_0406 | MSR0000_0407 | MSRC001_0045 |
| MC2 | MSR0000_0408 | MSR0000_0409 | MSR0000_040A | MSR0000_040B | MSRC001_0046 |
| MC3 | MSR0000_040C | MSR0000_040D | MSR0000_040E | MSR0000_040F | MSRC001_0047 |
| MC4 | MSR0000_0410 | MSR0000_0411 | MSR0000_0412 | MSR0000_0413 | MSRC001_0048 |
| MC5 | MSR0000_0414 | MSR0000_0415 | MSR0000_0416 | MSR0000_0417 | MSRC001_0049 |

Correctable and uncorrectable errors that are enabled in MCi_CTL are logged in MCi_STATUS and MCi_ADDR as they occur. Uncorrectable errors immediately result in a Machine Check exception.

Each MCi_CTL register must be enabled by the corresponding enable bit in [The Global Machine Check Exception Reporting Control Register (MCG_CTL)] MSR0000_017B.

Additionally, the MCi_CTL_MASK registers allow BIOS to mask the presence of any error source enables from software for test and debug. When error sources are masked, it is as if the error was not detected. Such masking consequently prevents error responses.

Each register bank implements a number of machine check miscellaneous registers, denoted as MCi_MISCj, where j goes from 0 to a maximum of 8. The presence of valid information in the first MCi_MISC register (MCi_MISC0) is indicated by MCi_STATUS[MiscV], and in subsequent registers by MCi_MISCj[Valid]. If there is more than one MCi_MISC register in a given bank, a non-zero value in MCi_MISC0[BlkPtr] points to the contiguous block of additional registers.

### 2.16.2 Machine Check Errors

There are two classes of machine check errors defined:
- Correctable: errors that can be corrected by hardware or microcode and cause no loss of data or corruption of processor state.
- Uncorrectable: errors that cannot be corrected by hardware or microcode and may have caused the loss of data or corruption of processor state.

Correctable errors are always corrected (unless disabled by implementation-specific bits in control registers for test or debug reasons). If they are enabled for logging, the status and address registers in the corresponding register bank are written with information that identifies the source of the error.

Uncorrectable errors, if enabled for logging, update the status and address registers, and if enabled for reporting, cause a machine check exception. If there is information in the status and address registers from a previous correctable error, it is overwritten. If an uncorrectable error is not enabled for logging, the error is ignored.

The implications of the two main categories of errors are (shown with a non-exhaustive list of examples):

1. Corrected error; the problem was dealt with.
   - Operationally (error handling), no action needs to be taken, because program flow is unaffected.
   - Diagnostically (fault management), software may collect information to determine if any components should be de-configured or serviced.
   - Examples include:
     - Correctable ECC, corrected online.
2. Uncorrected error; the problem was not dealt with.
   - Operationally (error handling), action does need to be taken, because program flow is affected.
   - Diagnostically (fault management), software may collect information to determine if and what components should be de-configured or serviced.
   - Examples include:
     - Uncorrectable ECC, no way to avoid passing it to process.

Machine check conditions can be simulated by using MSRC001_0015[McStatusWrEn]. This is useful for debugging machine check handlers.

#### 2.16.2.1 Machine Check Error Logging and Reporting

An error is considered enabled for logging if:
- The global enable for the corresponding error-reporting bank in [The Global Machine Check Exception Reporting Control Register (MCG_CTL)] MSR0000_017B is set to 1.
- The corresponding mask bit for the error in the MCi_CTL_MASK is cleared to 0.

An error is considered enabled for reporting if:
- The error is enabled for logging.
- The corresponding enable bit for the error in MC*i*_CTL is set to 1.

#### 2.16.2.2 Machine Check Error Logging Overwrite During Overflow

During error overflow conditions (see MSR0000_0401[Over] and MSR0000_0411[Over]), an error which has already been logged in the status register may be overwritten.

Table 42 indicates which errors are overwritten in the MC0 and MC4 error status registers. Table 43 indicates

which errors are overwritten in the MC1, MC2, MC3, and MC5 error status registers.

**Table 42: MC0 and MC4 overwrite priorities**

| | | | Older Error | | | |
|---|---|---|---|---|---|---|
| | | | Uncorrectable | | Correctable | |
| | | | Enabled | Disabled | Enabled | Disabled |
| Younger Error | Uncorrectable | Enabled | - | Overwrite | Overwrite | Overwrite |
| | | Disabled | - | Overwrite | Overwrite | Overwrite |
| | Correctable | Enabled | - | Overwrite | Overwrite | Overwrite |
| | | Disabled | - | Overwrite | Overwrite | Overwrite |

**Table 43: MC1, MC2, MC3, and MC5 overwrite priorities**

| | | | Older Error | | | |
|---|---|---|---|---|---|---|
| | | | Uncorrectable | | Correctable | |
| | | | Enabled | Disabled | Enabled | Disabled |
| Younger Error | Uncorrectable | Enabled | - | Overwrite | Overwrite | Overwrite |
| | | Disabled | - | Overwrite | Overwrite | Overwrite |
| | Correctable | Enabled | - | Overwrite | - | Overwrite |
| | | Disabled | - | Overwrite | - | Overwrite |

### 2.16.2.3 Machine Check Error Codes

The MCi_STATUS[ErrorCode] field contains information on the logged error. Table 44 identifies how to decode this field.

Three error code types are reported: TLB, memory, and bus errors. For a given error reporting bank, Error Code Type is used in conjunction with the Extended Error Code (MCi_STATUS[ErrorCodeExt]) to uniquely identify the Error Type. Details for each Error Type are described in the tables accompanying the MCi_STATUS register for each bank.

- MC0 (DC); Table 144
- MC1 (IC); Table 147
- MC2 (BU); Table 150
- MC3 (LS); Table 153
- MC4 (NB); Table 118 and Table 119
- MC5 (FR); Table 154

**Table 44: Error code types**

| Error Code | Error Code Type | Description |
|---|---|---|
| 0000 0000 0001 TTLL | TLB | Errors in the TLB cache.<br>TT = Transaction Type<br>LL = Cache Level |
| 0000 0001 RRRR TTLL | Memory | Errors in the cache hierarchy (not in NB)<br>RRRR = Memory Transaction Type<br>TT = Transaction Type<br>LL = Cache Level |
| 0000 1PPT RRRR IILL | Bus | General bus errors including link and DRAM<br>PP = Participation Processor<br>T = Timeout<br>RRRR = Memory Transaction Type<br>II = Memory or IO<br>LL = Cache Level |

**Table 45: Error codes: transaction type**

| TT | Transaction Type |
|---|---|
| 00 | Instr: Instruction |
| 01 | Data |
| 10 | Gen: Generic |
| 11 | Reserved |

**Table 46: Error codes: cache level**

| LL | Cache Level |
|---|---|
| 00 | Reserved |
| 01 | L1: Level 1 |
| 10 | L2: Level 2 |
| 11 | LG: Generic |

**Table 47: Error codes: memory transaction type**

| RRRR | Memory Transaction Type |
|---|---|
| 0000 | Gen: Generic. |
| 0001 | RD: Generic Read |
| 0010 | WR: Generic Write |
| 0011 | DRD: Data Read |
| 0100 | DWR: Data Write |
| 0101 | IRD: Instruction Fetch |

**Table 47: Error codes: memory transaction type**

| RRRR | Memory Transaction Type |
|---|---|
| 0110 | Prefetch |
| 0111 | Evict |
| 1000 | Snoop (Probe) |

**Table 48: Error codes: participation processor**

| PP | Participation Processor |
|---|---|
| 00 | SRC: Local node originated the request |
| 01 | RES: Local node responded to the request |
| 10 | OBS: Local node observed the error as a third party |
| 11 | Generic |

**Table 49: Error codes: memory or IO**

| II | Memory or IO |
|---|---|
| 00 | Mem: Memory Access |
| 01 | Reserved |
| 10 | IO: IO Access |
| 11 | Gen: Generic |

### 2.16.3 Handling Machine Check Exceptions

At a minimum, the machine check handler must be capable of logging errors for later examination. The handler should log as much information as is needed to diagnose the error.

More thorough exception handler implementations can analyze errors to determine if each error is recoverable. If a recoverable error is identified, the exception handler can attempt to correct the error and restart the interrupted program. Keep in mind that an error may not be recoverable for the process it directly affects, but may be containable to only that process, so that other processes in the system are unaffected.

Machine check exception handlers that attempt to recover must be thorough in their analysis and the corrective actions they take. The following guidelines should be used when writing such a handler:

- All status registers in the error-reporting banks must be examined to identify the cause of the machine check exception. Read [The Global Machine Check Capabilities Register (MCG_CAP)] MSR0000_0179[Count] to determine the number of status registers visible to each core. The status registers are numbered from 0 to one less than the value found in MSR0000_0179[Count]. For example, if the Count field indicates five status registers are supported, they are numbered MC0_STATUS to MC4_STATUS.
- Check the valid bit in each status register (MC*i*_STATUS[Val]). The remainder of the MC*i*_STATUS register does not need to be examined when its valid bit is clear.
- When identifying the error condition, portable exception handlers should examine MC*i*_STATUS[Error Code] and [ErrorCodeExt].
- When logging errors, particularly those that are not recoverable, check [The Global Machine Check Status Register (MCG_STAT)] MSR0000_017A[EIPV] to see if the instruction pointer address pushed onto the exception handler stack is related to the machine check. If EIPV is clear, the address is not guar-

anteed to be related to the error.

- Check the valid MC*i*_STATUS registers to see if error recovery is possible. Error recovery is not possible when:
  - The processor context corrupt indicator (MC*i*_STATUS[PCC]) is set to 1.
  - The error overflow status indicator (MC*i*_STATUS[Over]) is set to 1. This indicates that more than one machine check error has occurred, but only one error is reported by the status register.

  If error recovery is not possible, the handler should log the error information and return to the operating system.
- Check MC*i*_STATUS[UC] to see if the processor corrected the error. If UC is set, the processor did not correct the error, and the exception handler must correct the error prior to attempting to restart the interrupted program. If the handler cannot correct the error, it should log the error information and return to the operating system.
- If [The Global Machine Check Status Register (MCG_STAT)] MSR0000_017A[RIPV] is set, the interrupted program can be restarted reliably at the instruction pointer address pushed onto the exception handler stack. If RIPV is clear, the interrupted program cannot be restarted reliably, although it may be possible to restart it for debugging purposes.
- Prior to exiting the machine check handler, be sure to clear [The Global Machine Check Status Register (MCG_STAT)] MSR0000_017A[MCIP]. MCIP indicates that a machine check exception is in progress. If this bit is set when another machine check exception occurs, the processor enters the shutdown state.
- When an exception handler is able to successfully log an error condition, clear the MC*i*_STATUS registers prior to exiting the machine check handler. Software is responsible for clearing at least MC*i*_STATUS[Val].

Additional machine check handler portability can be added by having the handler use the CPUID instruction to identify the processor and its capabilities. Implementation specific software can be added to the machine check exception handler based on the processor information reported by CPUID.

#### 2.16.3.1 ECC

The syndrome field uniquely identifies the failing bit positions of a correctable ECC error. Only syndromes identified by Table 50 are correctable by the error correcting code.

Symbols 00h-0Fh map to data bits 0-63; symbols 10h-1Fh map to data bits 64-127; symbols 20-21h map to ECC check bits for data bits 0-63; symbols 22-23h map to ECC check bits for data bits 64-127.

To use Table 50, first find the 16-bit syndrome value in the table. This is most easily done by using low order 4 bits of the syndrome to select the appropriate error bitmask column. The entire four digit syndrome should then be in one of the rows of that column. The Symbol in error row indicates which symbol has the error, and the column indicates which bits within the symbol.

For example, if the ECC syndrome is 6913h, then symbol 05h has the error, and bits 0 and 1 within that symbol are corrupted, since the syndrome is in column 3h (0011b). Symbol 05h maps to bits 23-20, so the corrupted bits are 20 and 21.

**Table 50: ECC correctable syndromes**

| Symbol in error | Error bitmask | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 | 1000 | 1001 | 1010 | 1011 | 1100 | 1101 | 1110 | 1111 |
| Data 0 | e821 | 7c32 | 9413 | bb44 | 5365 | c776 | 2f57 | dd88 | 35a9 | a1ba | 499b | 66cc | 8eed | 1afe | f2df |
| Data 1 | 5d31 | a612 | fb23 | 9584 | c8b5 | 3396 | 6ea7 | eac8 | b7f9 | 4cda | 11eb | 7f4c | 227d | d95e | 846f |

**Table 50: ECC correctable syndromes**

| Symbol in error | Error bitmask | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 | 1000 | 1001 | 1010 | 1011 | 1100 | 1101 | 1110 | 1111 |
| Data 2 | 0001 | 0002 | 0003 | 0004 | 0005 | 0006 | 0007 | 0008 | 0009 | 000a | 000b | 000c | 000d | 000e | 000f |
| Data 3 | 2021 | 3032 | 1013 | 4044 | 6065 | 7076 | 5057 | 8088 | a0a9 | b0ba | 909b | c0cc | e0ed | f0fe | d0df |
| Data 4 | 5041 | a082 | f0c3 | 9054 | c015 | 30d6 | 6097 | e0a8 | b0e9 | 402a | 106b | 70fc | 20bd | d07e | 803f |
| Data 5 | be21 | d732 | 6913 | 2144 | 9f65 | f676 | 4857 | 3288 | 8ca9 | e5ba | 5b9b | 13cc | aded | c4fe | 7adf |
| Data 6 | 4951 | 8ea2 | c7f3 | 5394 | 1ac5 | dd36 | 9467 | a1e8 | e8b9 | 2f4a | 661b | f27c | bb2d | 7cde | 358f |
| Data 7 | 74e1 | 9872 | ec93 | d6b4 | a255 | 4ec6 | 3a27 | 6bd8 | 1f39 | f3aa | 874b | bd6c | c98d | 251e | 51ff |
| Data 8 | 15c1 | 2a42 | 3f83 | cef4 | db35 | e4b6 | f177 | 4758 | 5299 | 6d1a | 78db | 89ac | 9c6d | a3ee | b62f |
| Data 9 | 3d01 | 1602 | 2b03 | 8504 | b805 | 9306 | ae07 | ca08 | f709 | dc0a | e10b | 4f0c | 720d | 590e | 640f |
| Data 10 | 9801 | ec02 | 7403 | 6b04 | f305 | 8706 | 1f07 | bd08 | 2509 | 510a | c90b | d60c | 4e0d | 3a0e | a20f |
| Data 11 | d131 | 6212 | b323 | 3884 | e9b5 | 5a96 | 8ba7 | 1cc8 | cdf9 | 7eda | afeb | 244c | f57d | 465e | 976f |
| Data 12 | e1d1 | 7262 | 93b3 | b834 | 59e5 | ca56 | 2b87 | dc18 | 3dc9 | ae7a | 4fab | 642c | 85fd | 164e | f79f |
| Data 13 | 6051 | b0a2 | d0f3 | 1094 | 70c5 | a036 | c067 | 20e8 | 40b9 | 904a | f01b | 307c | 502d | 80de | e08f |
| Data 14 | a4c1 | f842 | 5c83 | e6f4 | 4235 | 1eb6 | ba77 | 7b58 | df99 | 831a | 27db | 9dac | 396d | 65ee | c12f |
| Data 15 | 11c1 | 2242 | 3383 | c8f4 | d935 | eab6 | fb77 | 4c58 | 5d99 | 6e1a | 7fdb | 84ac | 956d | a6ee | b72f |
| Data 16 | 45d1 | 8a62 | cfb3 | 5e34 | 1be5 | d456 | 9187 | a718 | e2c9 | 2d7a | 68ab | f92c | bcfd | 734e | 369f |
| Data 17 | 63e1 | b172 | d293 | 14b4 | 7755 | a5c6 | c627 | 28d8 | 4b39 | 99aa | fa4b | 3c6c | 5f8d | 8d1e | eeff |
| Data 18 | b741 | d982 | 6ec3 | 2254 | 9515 | fbd6 | 4c97 | 33a8 | 84e9 | ea2a | 5d6b | 11fc | a6bd | c87e | 7f3f |
| Data 19 | dd41 | 6682 | bbc3 | 3554 | e815 | 53d6 | 8e97 | 1aa8 | c7e9 | 7c2a | a16b | 2ffc | f2bd | 497e | 943f |
| Data 20 | 2bd1 | 3d62 | 16b3 | 4f34 | 64e5 | 7256 | 5987 | 8518 | aec9 | b87a | 93ab | ca2c | e1fd | f74e | dc9f |
| Data 21 | 83c1 | c142 | 4283 | a4f4 | 2735 | 65b6 | e677 | f858 | 7b99 | 391a | badb | 5cac | df6d | 9dee | 1e2f |
| Data 22 | 8fd1 | c562 | 4ab3 | a934 | 26e5 | 6c56 | e387 | fe18 | 71c9 | 3b7a | b4ab | 572c | d8fd | 924e | 1d9f |
| Data 23 | 4791 | 89e2 | ce73 | 5264 | 15f5 | db86 | 9c17 | a3b8 | e429 | 2a5a | 6dcb | f1dc | b64d | 783e | 3faf |
| Data 24 | 5781 | a9c2 | fe43 | 92a4 | c525 | 3b66 | 6ce7 | e3f8 | b479 | 4a3a | 1dbb | 715c | 26dd | d89e | 8f1f |
| Data 25 | bf41 | d582 | 6ac3 | 2954 | 9615 | fcd6 | 4397 | 3ea8 | 81e9 | eb2a | 546b | 17fc | a8bd | c27e | 7d3f |
| Data 26 | 9391 | e1e2 | 7273 | 6464 | f7f5 | 8586 | 1617 | b8b8 | 2b29 | 595a | cacb | dcdc | 4f4d | 3d3e | aeaf |
| Data 27 | cce1 | 4472 | 8893 | fdb4 | 3155 | b9c6 | 7527 | 56d8 | 9a39 | 12aa | de4b | ab6c | 678d | ef1e | 23ff |
| Data 28 | a761 | f9b2 | 5ed3 | e214 | 4575 | 1ba6 | bcc7 | 7328 | d449 | 8a9a | 2dfb | 913c | 365d | 688e | cfef |
| Data 29 | ff61 | 55b2 | aad3 | 7914 | 8675 | 2ca6 | d3c7 | 9e28 | 6149 | cb9a | 34fb | e73c | 185d | b28e | 4def |
| Data 30 | 5451 | a8a2 | fcf3 | 9694 | c2c5 | 3e36 | 6a67 | ebe8 | bfb9 | 434a | 171b | 7d7c | 292d | d5de | 818f |
| Data 31 | 6fc1 | b542 | da83 | 19f4 | 7635 | acb6 | c377 | 2e58 | 4199 | 9b1a | f4db | 37ac | 586d | 82ee | ed2f |
| Check0 | be01 | d702 | 6903 | 2104 | 9f05 | f606 | 4807 | 3208 | 8c09 | e50a | 5b0b | 130c | ad0d | c40e | 7a0f |
| Check1 | 4101 | 8202 | c303 | 5804 | 1905 | da06 | 9b07 | ac08 | ed09 | 2e0a | 6f0b | f40c | b50d | 760e | 370f |
| Check2 | c441 | 4882 | 8cc3 | f654 | 3215 | bed6 | 7a97 | 5ba8 | 9fe9 | 132a | d76b | adfc | 69bd | e57e | 213f |
| Check3 | 7621 | 9b32 | ed13 | da44 | ac65 | 4176 | 3757 | 6f88 | 19a9 | f4ba | 829b | b5cc | c3ed | 2efe | 58df |

## 2.17 Sideband Interface (SBI)

The sideband interface (SBI) is an SMBus v2.0 compatible 2-wire processor slave interface. SBI is also referred as the Advanced Platform Management Link.

SBI is used to communicate with the Temperature Sensor Interface (SB-TSI) (see the *SBI Temperature Sensor Interface (SB-TSI) Specification*, #40821) and 2.10.2 [Sideband Temperature Sensor Interface (SB-TSI)]).

### 2.17.1 SBI Processor Information

Processor access to the SBI configuration is via [The SBI Control Register] D18F3x1E4. The processor can access SB-TSI registers through [The SBI Address Register] D18F3x1E8 and [The SBI Data Register] D18F3x1EC.

# 3 Registers

This section provides detailed field definitions for the register sets in the processor.

## 3.1 Register Descriptions and Mnemonics

Each register in this document is referenced with a mnemonic. Each mnemonic is a concatenation of the register-space indicator and the offset of the register. Here are the mnemonics for the various register spaces:

- **IOXXX**: x86-defined input and output address space registers; XXX specifies the hexadecimal byte address of the IO instruction. This space includes IO-space configuration access registers [The IO-Space Configuration Address Register] IOCF8 and [The IO-Space Configuration Data Port] IOCFC.
- **APICXX0**: APIC memory-mapped registers; XX0 is the hexadecimal byte address offset from the base address. The base address for this space is specified by [The APIC Base Address Register (APIC_BAR)] MSR0000_001B.
- **CPUID FnXXXX_XXXX**: processor capabilities information returned by the CPUID instruction. See 3.19 [CPUID Instruction Registers]. There is one set of these registers per core; each core may only access its own set of these capabilities
- **MSRXXXX_XXXX**: model specific registers; XXXX_XXXX is the hexadecimal MSR number. This space is accessed through x86-defined RDMSR and WRMSR instructions. There is one set of these registers per core; each core may only access its own set of these registers.
- **DZFYxXXX**: PCI-defined configuration space; XXX specifies the hexadecimal byte address of the configuration register (this may be 2 or 3 digits); Y specifies the function number; Z defined the device number; e.g., D0F3x40 specifies the register of device 0 at function 3, address 40. See 2.7 [Configuration Space], for details about configuration space.
- **GMMxX_XXXX**: GPU memory mapped registers; X_XXXX specifies the hexadecimal byte address offset (this may be 2 to 5 digits) from the base address register; The base address for this space is specified by [The Graphics Memory Mapped Registers Base Address] D1F0x18.
- **PMCxXXX**: performance monitor events; XXX is the hexadecimal event counter number programmed into [The Performance Event Select Register (PERF_CTL[3:0])] MSRC001_00[03:00][EventSelect].
- **SMUxXX**: Internal SMU registers; XX specifies the hexadecimal byte address. See 3.15 [Internal System Management Unit (SMU) Registers] for details about SMU space.
- **FCRxXXXX_XXXX**: Fixed configuration registers used for various aspects of processor initialization; XXXX_XXXX is the hexadecimal address. See 3.16 [Fixed Configuration Space (FCR)] for details about FCR space.

Each mnemonic may specify the location of one or more registers that share the same base definition. A mnemonic that specifies more than one register will contain one or more ranges within braces. The ranges are specified as follows:

- Comma separated lists [A,B]: Define specific instances of a register, e.g., D0F3x[1,0]40 defines two registers D0F3x40 and D0F3x140.
- Colon separated ranges [A:B]: Defines all registers that contain the range between A and B. Examples:
  - D0F3x[50:40] defines five registers D0F3x40, D0F3x44, D0F3x48, D0F3x4C, and D0F3x50.
  - D[8:2]F0x40 defines seven registers D2F0x40, D3F0x40, D4F0x40, D5F0x40, D6F0x40, D7F0x40, and D8F0x40.
  - D0F0xE4_x013[2:0]_0000 defines three registers D0F0xE4_x0130_0000, D0F0xE4_x0131_0000, and D0F0xE4_x0132_0000.

- Colon separated ranges with a explicit step [A:B:stepC]: Defines the registers from B to A, C defines the offset between registers., e.g., D0F3x[50:40:step8] defines three registers D0F3x40, D0F3x48, and D0F3x50.

The processor includes a single set of IO-space and configuration-space registers. However, APIC, CPUID, and MSR register spaces are implemented once per core. Access to IO-space and configuration space registers may require software-level techniques to ensure that no more than one core attempts to access a register at a time.

The following is terminology found in the register descriptions.

**Table 51: Terminology in register descriptions**

| Terminology | Description |
|---|---|
| BIOS: | The recommended value to be set by software, either BIOS, OS, or driver. The format is defined to be BIOS: <integer-expression>.<br>• If a software recommendation does not exist for all conditions or for all bits of a condition, then software is recommended not to change the value of the specified bit(s).<br>• If "BIOS:" occurs in a register field:<br>  • The recommended value is applied to the field.<br>• If "BIOS:" occurs after a register name but outside of a register field table row:<br>  • The recommended value is applied to the width of the register.<br>• E.g.: BIOS: 4h.<br>• E.g.: BIOS: 1h if (F0x84[IsocEn]==1 && F0x68[DispRefModeEn]==0 && MSRC001_001F[EnConvertToNonIsoc]==0).<br>• |
| Access types | |
| Read | Capable of being read by software. |
| Read-only | Capable of being read but not written by software. |
| Write | Capable of being written by software. |
| Read-write | Capable of being written by software and read by software. |
| Write-only. | Capable of being written by software. Reads return undefined values. |
| Set-by-hardware, cleared-by-hardware, updated-by-hardware | Register bit is set or cleared by hardware. Register bit or field is updated by hardware. |
| Set-when-done, Cleared-when-done | Register bit is set or cleared by hardware when the operation completes. |
| Write-once | After RESET_L is asserted, these registers may be written to once. After being written, they become read-only until the next RESET_L assertion. The write-once control is byte based. So, for example, software may write each byte of a write-once DWORD as four individual transactions. As each byte is written, that byte becomes read-only. |
| Write-1-to-clear | Software must write a 1 to the bit in order to clear it. Writing a 0 to these bits has no affect. |
| Write-1-only | Software can set the bit high by writing a 1 to it. Writes of 0 have no effect. Cleared by hardware. |
| Reset-applied | Takes effect on warm reset. |
| Strap | Use strap configuration procedure to update. See 2.11.4.3.1 [Straps]. |
| Field definitions | |

**Table 51: Terminology in register descriptions**

| Terminology | Description |
|---|---|
| Reserved | Field is reserved for future use. Software is required to preserve the state read from these bits when writing to the register. Software may not depend on the state of reserved fields nor on the ability of such fields to return the state previously written. |
| Unused | Field is reserved for future use. Software is not required to preserve the state read from these bits when writing to the register. Software may not depend on the state of unused fields nor on the ability of such fields to return the state previously written. |
| MBZ | Must be zero. If software attempts to set an MBZ bit to 1, a general-protection exception (#GP) occurs. |
| RAZ | Read as zero. Writes are ignored, unless RAZ is combined with write, write-1-only, or write-once. |
| Reset definitions | |
| Reset | The reset value of each register is provided below the mnemonic or in the field description. Unless otherwise noted, the register state matches the reset value when RESET_L is asserted (either a cold or a warm reset). Reset values may include:<br>**?**: a question mark in the reset value indicates that the reader should look at the bit description for reset-value details.<br>**X**: an X in the reset value indicates that the field resets (warm or cold) to an unspecified state. |
| Cold reset | The field state is not affected by a warm reset (even if the field is labeled “cold reset: X”); it is placed into the reset state when PWROK is deasserted. See "Reset" above for the definition of characters that may be found in the cold reset value. |
| Value | The current value of a read-only field or register. A value statement explicitly defines the value returned under all conditions including after reset events. A field labeled “Value:” will not have a separate reset definition. |

### 3.1.1 Northbridge MSRs In Multi-Core Products

MSRs that control Northbridge functions are shared between all cores in a multi-core processor (e.g. MSR0000_0410). If control of Northbridge functions is shared between software on all cores, software must ensure that only one core at a time is allowed to access the shared MSR.

### 3.1.2 Software Recommendation (BIOS, SBIOS, CBIOS, etc.)

The following keywords specify the recommended value to be set by software.

- BIOS: AMD BIOS.
- SBIOS: Platform BIOS.
- CBIOS: Chip-set BIOS.
- OS: Operating system.
- Driver: Device driver software settings invoked by the OS.

Syntax: BIOS:<integer-expression>. Any of the supported tags can be substituted for BIOS.

If “BIOS:” occurs in a register field, then the recommended value is applied to the field. If “BIOS:” occurs after a register name but outside of a register field table row, then the recommended value is applied to the width of the register.

## 3.2 IO Space Registers

See 3.1 [Register Descriptions and Mnemonics] for a description of the register naming convention.

### IOCF8 IO-Space Configuration Address Register

Reset: 0000_0000h.

[The IO-Space Configuration Address Register] IOCF8, and [The IO-Space Configuration Data Port] IOCFC, are used to access system configuration space, as defined by the PCI specification. IOCF8 provides the address register and IOCFC provides the data port. Software sets up the configuration address by writing to IOCF8. Then, when an access is made to IOCFC, the processor generates the corresponding configuration access to the address specified in IOCF8. See 2.7 [Configuration Space].
IOCF8 may only be accessed through aligned, DW IO reads and writes; otherwise, the accesses are passed to the appropriate link. Accesses to IOCF8 and IOCFC received from a link are treated as all other IO transactions received from a link and are forwarded based on the settings in D18F1xC0 [IO-Space Base Register]. IOCF8 and IOCFC in the processor are not accessible from a link.

| Bits | Description |
|---|---|
| 31 | **ConfigEn: configuration space enable**. Read-write. 1=IO read and write accesses to IOCFC are translated into configuration cycles at the configuration address specified by this register. 0=IO read and write accesses are passed to the appropriate link and no configuration access is generated. |
| 30:28 | Reserved. |
| 27:24 | **ExtRegNo: extended register number**. Read-write. ExtRegNo provides bits[11:8] and RegNo provides bits[7:2] of the byte address of the configuration register. ExtRegNo is reserved unless it is enabled by MSRC001_001F[EnableCf8ExtCfg]. |
| 23:16 | **BusNo: bus number**. Read-write. Specifies the bus number of the configuration cycle. |
| 15:11 | **Device: device number**. Read-write. Specifies the device number of the configuration cycle. |
| 10:8 | **Function**. Read-write. Specifies the function number of the configuration cycle. |
| 7:2 | **RegNo: register address**. Read-write. See IOCF8[ExtRegNo]. |
| 1:0 | Reserved. |

### IOCFC IO-Space Configuration Data Port

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:0 | **ConfigData**. Read-write. See IOCF8 for details about this port. |

## 3.3 Device 0 Function 0 (Root Complex) Configuration Registers

See 3.1 [Register Descriptions and Mnemonics] for a description of the register naming convention. See 2.7 [Configuration Space] for details about how to access this space.

### D0F0x00 Device/Vendor ID Register

Reset: 1705_1022h.

| Bits | Description |
|---|---|
| 31:16 | **DeviceID: device ID**. Read-only. |
| 15:0 | **VendorID: vendor ID**. Read-only. |

**D0F0x04 Status/Command Register**

Reset: 0220_0004h.

| Bits | Description |
|---|---|
| 31 | **ParityErrorDetected: parity error detected**. Read-only. |
| 30 | **SignaledSystemError: signaled system error**. Read; write-1-to-clear. 1=FCH generated a system error. |
| 29 | **ReceivedMasterAbort: received master abort**. Read; write-1-to-clear. |
| 28 | **ReceivedTargetAbort: received target abort**. Read; write-1-to-clear. |
| 27 | **SignalTargetAbort: signaled target abort**. Read-only. |
| 26:25 | **DevselTiming: DEVSEL# Timing**. Read-only. |
| 24 | Reserved. |
| 23 | **FastBackCapable: fast back-to-back capable**. Read-only. |
| 22 | Reserved. |
| 21 | **PCI66En: 66 MHz capable**. Read-only. |
| 20 | **CapList: capability list**. Read-only. 1=Capability list supported. |
| 19:10 | Reserved. |
| 9 | **FastB2BEn: fast back-to-back enable**. Read-write. |
| 8 | **SerrEn: system error enable**. Read-write. |
| 7 | Reserved. |
| 6 | **ParityErrorEn: parity error response enable**. Read-only. |
| 5 | **PalSnoopEn: VGA palette snoop enable**. Read-only. |
| 4 | **MemWriteInvalidateEn: memory write and invalidate enable**. Read-only. |
| 3 | **SpecialCycleEn: special cycle enable**. Read-only. |
| 2 | **BusMasterEn: bus master enable**. Read-only. |
| 1 | **MemAccessEn: memory access enable**. Read-write. BIOS: 1. This bit controls if memory accesses by this device are accepted or not. 1=Enabled. 0=Disabled. |
| 0 | **IoAccessEn: IO access enable**. Read-only. |

**D0F0x08 Class Code/Revision ID Register**

Reset: 0600_00xxh.

| Bits | Description |
|---|---|
| 31:8 | **ClassCode: class code**. Read-only. Provides the host bridge class code as defined in the PCI specification. |
| 7:0 | **RevID: revision ID**. Read-only. |

**D0F0x0C Header Type Register**

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:24 | **BIST**. Read-only. |
| 23:16 | **HeaderTypeReg**. Read-only. 00h=Single function device. |
| 15:8 | **LatencyTimer**. Read-write. |
| 7:0 | **CacheLineSize**. Read-only. |

**D0F0x2C Subsystem and Subvendor ID**

| Bits | Description |
|---|---|
| 31:16 | **SubsystemID**. Value: 1705h. |
| 15:0 | **SubsystemVendorID**. Value: 1022h. |

**D0F0x34 Capabilities Pointer Register**

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:8 | Reserved. |
| 7:0 | **CapPtr: capabilities pointer**. Read-only. There is no capability list. |

**D0F0x4C PCI Control Register**

Reset: 0000_2002h.

| Bits | Description |
|---|---|
| 31:27 | Reserved. |
| 26 | **HPDis: hot plug message disable**. Read-write. BIOS: 0. 1=Hot plug message generation is disabled. |
| 25:24 | Reserved. |
| 23 | **MMIOEnable: memory mapped IO enable**. Read-write. BIOS: 0. 1=Decoding of MMIO cycles is enabled. |
| 22:15 | Reserved. |
| 14:12 | **CfgRdTime**. Read-write. BIOS: 010b. Specifies the propagation delay for read data on the configuration bus.<br>Bits Definition<br>111b-000b <7-CfgRdTime> Clocks. |
| 11 | **CRS**: **configuration request retry detected**. Read; Write-1-to-clear; set-by-hardware. 1=Configuration request retry was detected. |
| 10:6 | Reserved. |
| 5 | **SerrDis**: s**ystem error message disable**. Read-write. BIOS: 0. 1=The generation of SERR messages is disabled. |

| | |
|---|---|
| 4 | **PMEDis: PME disable**. Read-write. BIOS: 0. 1=The generation of PME messages is disabled. |
| 3 | **Cf8Dis: Cf8 disable**. Read-write. BIOS: 0. 1=Configuration accesses through IO address Cf8h to this device are disabled. |
| 2 | Reserved. |
| 1 | **ApicEnable: APIC enable**. Read-write. BIOS: 1. 1=APIC is enabled. |
| 0 | **Function1Enable: device 0 function 1 enable**. Read-write. BIOS: 0. 1=Configuration accesses to device 0 function 1 are enabled. |

#### D0F0x60 Miscellaneous Index Register

Reset: 0000_0000h.

The index/data pair registers D0F0x60 and D0F0x64 is used to access the registers D0F0x64_x[FF:00]. To read or write to one of these register, the address is written first into the address register D0F0x60 and then the data are read or written by read or write the data register D0F0x64.

| Bits | Description |
|---|---|
| 31:8 | Reserved. |
| 7 | **MiscIndWrEn: miscellaneous index write enable**. Read-write. If set writes to D0F0x64 are enabled. |
| 6:0 | **MiscIndAddr: miscellaneous index register address**. Read-write. |

#### D0F0x64 Miscellaneous Index Data Register

See D0F0x60.

#### D0F0x64_x00 Northbridge Control

Reset: 0000_0002h.

| Bits | Description |
|---|---|
| 31:8 | Reserved. |
| 7 | **HwInitWrLock**. Read-write. BIOS: 1. 1=Lock HWInit registers. 0=Unlock HWInit registers. |
| 6 | **NbFchCfgEn: device 8 enable**. Read-write. BIOS: 0. 1=Bridge device 8 is enabled. |
| 5:0 | Reserved. |

#### D0F0x64_x0B IOC Link Control

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:24 | Reserved. |
| 23 | **IocFchSetPmeTurnOffEn**. Read-write. BIOS: 0. 1=Enables the PME_Turn_Off/PME_To_Ack mechanism between northbridge and FCH. |
| 22 | Reserved. |
| 21 | **IocFchSetPowEn: set slot power enable**. Read-write. BIOS: 0. 1=Enables sending set_slot_power_limit/scale messages to the FCH. |

| | |
|---|---|
| 20 | **SetPowEn: set slot power enable**. Read-write. BIOS: 0. 1=Enables sending set_slot_power messages to the FCH. |
| 19:0 | Reserved. |

**D0F0x64_x0C IOC Bridge Control**

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:8 | Reserved. |
| 7 | **Dev7BridgeDis**. Read-write. 1=Bus 0, device 7 bridge functionality is hidden. |
| 6 | **Dev6BridgeDis**. Read-write. 1=Bus 0, device 6 bridge functionality is hidden. |
| 5 | **Dev5BridgeDis**. Read-write. 1=Bus 0, device 5 bridge functionality is hidden. |
| 4 | **Dev4BridgeDis**. Read-write.1=Bus 0, device 4 bridge functionality is hidden. |
| 3 | **Dev3BridgeDis**. Read-write.1=Bus 0, device 3 bridge functionality is hidden. |
| 2 | **Dev2BridgeDis**. Read-write.1=Bus 0, device 2 bridge functionality is hidden. |
| 1:0 | Reserved. |

**D0F0x64_x19 Top of Memory 2 Low**

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:23 | **Tom2[31:23]: top of memory 2**. Read-write. BIOS: MSRC001_001D[Tom2[31:23]]. This field specifies the maximum system address for upstream read and write transactions that are forwarded to the host bridge. All addresses less than or equal to this system address are forwarded to DRAM and are not checked to determine if the transaction is a peer-to-peer transaction. All upstream reads with addresses greater than this system address are master aborted. |
| 22:1 | Reserved. |
| 0 | **TomEn: top of memory enable**. Read-write. BIOS: MSRC001_0010[MtrrTom2En]. 1=Top of memory check enabled. |

**D0F0x64_x1A Top of Memory 2 High**

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:8 | Reserved. |
| 7:0 | **Tom2[39:32]: top of memory 2**. Read-write. BIOS: MSRC001_001D[Tom2[39:32]]. See D0F0x64_x19[Tom2]. |

**D0F0x64_x1C Internal Graphics PCI Control 1**

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:24 | Reserved. |
| 23 | **RcieEn**. IF (D0F0x64_x1C[WriteDis]==1) THEN Read-only. ELSE Read-write. ENDIF. BIOS: 1. 1=Root complex integrated endpoint mode. 0=Legacy PCI device mode. |
| 22 | Reserved. |
| 21 | **PcieDis**. IF (D0F0x64_x1C[WriteDis]==1) THEN Read-only. ELSE Read-write. ENDIF. BIOS: 0. |
| 20:18 | Reserved. |
| 17 | **F0En**. IF (D0F0x64_x1C[WriteDis]==1) THEN Read-only. ELSE Read-write. ENDIF. BIOS:1. 1=Internal graphics enabled. 0=Internal graphics disabled. |
| 16 | **IoBarDis**. IF (D0F0x64_x1C[WriteDis]==1) THEN Read-only. ELSE Read-write. ENDIF. BIOS: 0 1=Graphics IO base address register disabled. 0=Graphics IO base address register enabled. |
| 15:12 | Reserved. |
| 11 | **Audio64BarEn**. IF (D0F0x64_x1C[WriteDis]==1) THEN Read-only. ELSE Read-write. ENDIF. Controls the size of the audio BAR. 1=64-bit BAR. 0=32-bit BAR. |
| 10 | **AudioNonlegacyDeviceTypeEn**. IF (D0F0x64_x1C[WriteDis]==1) THEN Read-only. ELSE Read-write. ENDIF. BIOS: 0. 1=PCIe device. 0=Legacy PCI device. |
| 9 | **MsiDis**. IF (D0F0x64_x1C[WriteDis]==1) THEN Read-only. ELSE Read-write. ENDIF. BIOS: 0. 1=MSI interrupts disabled. 0=MSI interrupts enabled. |
| 8 | **AudioEn**. IF (D0F0x64_x1C[WriteDis]==1) THEN Read-only. ELSE Read-write. ENDIF. BIOS: 1. 1=HDMI audio enabled. 0=HDMI audio disabled. |
| 7 | Reserved. |
| 6 | **RegApSize**. IF (D0F0x64_x1C[WriteDis]==1) THEN Read-only. ELSE Read-write. ENDIF. BIOS: 1. Specifies the size of the graphics register aperture. 0=64KB. 1=256KB. |
| 5:3 | **MemApSize**. IF (D0F0x64_x1C[WriteDis]==1) THEN Read-only. ELSE Read-write. ENDIF. Specifies the size of the frame buffer aperture.<br>Bits Definition Bits Definition<br>000b 128MB 100b 512MB<br>001b 256MB 101b 1GB<br>010b 64MB 110b 2GB<br>011b 32MB 111b 4GB |
| 2 | **F064BarEn**. IF (D0F0x64_x1C[WriteDis]==1) THEN Read-only. ELSE Read-write. ENDIF. 1=64-bit base address registers. 0=32-bit base address registers. |
| 1 | **F0NonlegacyDeviceTypeEn**. IF (D0F0x64_x1C[WriteDis]==1) THEN Read-only. ELSE Read-write. ENDIF. BIOS: 0. 1=PCIe device. 0=Legacy PCI device. |
| 0 | **WriteDis**. IF (D0F0x64_x1C[WriteDis]==1) THEN Read-only. ELSE Read-write. ENDIF. BIOS: 1. 1=Register is read-only and the GPU PCI interface is configured. 0=Register is read-write. See 2.14.1 [GPU PCI Interface]. |

**D0F0x64_x1D Internal Graphics PCI Control 2**

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:4 | Reserved. |
| 3 | **Vga16En: VGA IO 16 bit decoding enable**. Read-write. BIOS: 1. 1=Address bits 15:10 for VGA IO cycles are decoded. 0=Address bits 15:10 for VGA IO cycles are ignored. |
| 2 | Reserved. |
| 1 | **VgaEn: VGA enable**. Read-write. BIOS: 1. Affects the response by the internal graphics to compatible VGA addresses when IntGfxAsPcieEn=1. 1=The internal graphics decodes the following accesses:<br>• Memory accesses in the range of A0000h to BFFFFh.<br>• IO address where address bits 9:0 are in the ranges of 3B0h to 3BBh or 3C0h to 3DFh. For IO cycles the decoding of address bits 15:10 depends on Vga16En. |
| 0 | **IntGfxAsPcieEn: internal graphics is RC integrated device**. Read-write. BIOS: 1. 1=Integrated graphics is device 1 on bus 0 and operates as a RC integrated device. Software must program D1F0x04[BusMasterEn]=1 before programming IntGfxAsPcieEn=1. 0=Integrated graphics is located behind a PCI-to-PCI bridge and is device 5 on the bus behind the bridge. The bridge device is device 1 on bus 0. |

**D0F0x64_x20 Device Remap Register**

Reset: 0000_0002h.

| Bits | Description |
|---|---|
| 31:2 | Reserved. |
| 1 | **IocPcieDevRemapDis**. Read-write. 0=GPP bridge devices numbers are assigned sequentially. 1=GPP bridge device numbers are assigned based on the first lane of the link. See Table 36 [Supported General Purpose (GPP) Link Configurations]. |
| 0 | Reserved. |

**D0F0x64_x22 LCLK Control 0**

Reset: 7F3F_8100h. See 2.5.5.3 [GPU and Root Complex Clock Gating].

| Bits | Description |
|---|---|
| 31 | Reserved. |
| 30 | **SoftOverrideClk0**. Read-write. BIOS: 0. 1=Dynamic clock gating disabled for the host request path to the PCIe cores. |
| 29 | **SoftOverrideClk1**. Read-write. BIOS: 0. 1=Dynamic clock gating disabled for the host request path to the internal graphics and the host response path. |
| 28 | **SoftOverrideClk2**. Read-write. BIOS: 0. 1=Dynamic clock gating disabled for the host configuration requests. |
| 27 | **SoftOverrideClk3**. Read-write. BIOS: 0. 1=Dynamic clock gating disabled for the debug bus path. |

| | |
|---|---|
| 26 | **SoftOverrideClk4**. Read-write. BIOS: 0. 1=Dynamic clock gating disabled for the host request path to the configuration block. |
| 25:12 | Reserved. |
| 11:4 | **OffHysteresis**. Read-write. Specifies the number of LCLK cycles hardware waits after logic goes idle before gating the clocks specifies by SoftOverrideClk[4:0]. |
| 3:0 | Reserved. |

**D0F0x64_x23 LCLK Control 1**

Reset: 7F3F_8100h. See 2.5.5.3 [GPU and Root Complex Clock Gating].

| Bits | Description |
|---|---|
| 31 | Reserved. |
| 30 | **SoftOverrideClk0**. Read-write. BIOS: 0. 1=Dynamic clock gating disabled for upstream DMA requests from all sources. |
| 29 | **SoftOverrideClk1**. Read-write. BIOS: 0. 1=Dynamic clock gating disabled for upstream DMA requests from the GPPFCH link core. |
| 28 | **SoftOverrideClk2**. Read-write. BIOS: 0. 1=Dynamic clock gating disabled for upstream DMA requests from internal graphics and its DMA response reordering path. |
| 27 | **SoftOverrideClk3**. Read-write. BIOS: 0. 1=Dynamic clock gating disabled for upstream DMA requests from internal graphics. |
| 26 | **SoftOverrideClk4**. Read-write. BIOS: 0. 1=Dynamic clock gating disabled for upstream DMA requests from the Gfx link core. |
| 25:12 | Reserved. |
| 11:4 | **OffHysteresis**. Read-write. Specifies the number of LCLK cycles hardware waits after logic goes idle before gating the clocks specifies by SoftOverrideClk[4:0]. |
| 3:0 | Reserved. |

**D0F0x64_x24 SCLK Control**

Reset: 783C_0100h. See 2.5.5.3 [GPU and Root Complex Clock Gating].

| Bits | Description |
|---|---|
| 31 | Reserved. |
| 30 | **SoftOverrideClk0**. Read-write. BIOS: 0. 1=Dynamic clock gating disabled for host request to internal graphics. |
| 29 | **SoftOverrideClk1**. Read-write. BIOS: 0. 1=Dynamic clock gating disabled for upstream DMA requests from internal graphics. |
| 28:12 | Reserved. |
| 11:4 | **OffHysteresis**. Read-write. Specifies the number of SCLK cycles hardware waits after logic goes idle before gating the clocks specifies by SoftOverrideClk[1:0]. |
| 3:0 | Reserved. |

### D0F0x64_x46 IOC Features Control

Reset: 0000_3063h.

| Bits | Description |
|---|---|
| 31:17 | Reserved. |
| 16 | **Msi64bitEn: 64 bit MSI enable**. Read-write. BIOS: 1. 1=64-bit MSI support enabled. 0=64-bit MSI support disabled. |
| 15:3 | Reserved. |
| 2:1 | **P2PMode: peer-to-peer mode**. Read-write. BIOS: 00b. Specifies how upstream write transactions above D0F0x64_x19[Tom2] are completed.<br>Bits — Definition<br>00b — Master abort writes that do not hit one of the internal PCI bridges. Forward writes that hit one of the internal PCI bridges to the bridge.<br>11b-01b — Reserved. |
| 0 | Reserved. |

### D0F0x64_x4D SMU Request Port

Reset: 0000_0000h. See 3.15 [Internal System Management Unit (SMU) Registers].

| Bits | Description |
|---|---|
| 31:26 | Reserved. |
| 25 | **ReqType**. Read-write. 1=Write. 0=Read. |
| 24 | **ReqToggle: request toggle**. Read-write. |
| 23:16 | **SmuAddr**. Read-write. Specifies the SMU register address. |
| 15:0 | **WriteData**. Read-write. Specifies the data written to the SMU. |

### D0F0x64_x4E SMU Read Data

Reset: 0000_0000h. See 3.15 [Internal System Management Unit (SMU) Registers].

| Bits | Description |
|---|---|
| 31:0 | **SmuReadData**. Read-only. Returns the data read from the SMU. |

### D0F0x64_x5[B,9,7,5,3,1] IOC PCIe Device Control

Reset: 0000_0000h.

**Table 52: Index address mapping for** D0F0x64_x5[B,9,7,5,3,1]

| D0F0x64 | Function | BIOS |
|---|---|---|
| 51h | Device 2 | IF (D0F0x64_x0C[Dev2BridgeDis]==0) THEN 0010_0000h ELSE 0000_0000h ENDIF. |
| 53h | Device 3 | IF (D0F0x64_x0C[Dev3BridgeDis]==0) THEN 0010_0000h ELSE 0000_0000h ENDIF. |
| 55h | Device 4 | IF (D0F0x64_x0C[Dev4BridgeDis]==0) THEN 0010_0000h ELSE 0000_0000h ENDIF. |

**Table 52: Index address mapping for** D0F0x64_x5[B,9,7,5,3,1]

| | | |
|---|---|---|
| 57h | Device 5 | IF (D0F0x64_x0C[Dev5BridgeDis]==0) THEN 0010_0000h ELSE 0000_0000h ENDIF. |
| 59h | Device 6 | IF (D0F0x64_x0C[Dev6BridgeDis]==0) THEN 0010_0000h ELSE 0000_0000h ENDIF. |
| 5Bh | Device 7 | IF (D0F0x64_x0C[Dev7BridgeDis]==0) THEN 0010_0000h ELSE 0000_0000h ENDIF. |

| Bits | Description |
|---|---|
| 31:21 | Reserved. |
| 20 | **SetPowEn: set slot power enable**. Read-write. BIOS: 1. 1=Enables the set_slot_power message to the FCH. |
| 19:0 | Reserved. |

### D0F0x64_x6A Voltage Control

Reset: 0000_0000h. See 2.5.1.5.2 [Software-Initiated Voltage Transitions].

| Bits | Description |
|---|---|
| 31:5 | Reserved. |
| 4:3 | **VoltageLevel**. Read-write. Specifies the VID requested when software toggles D0F0x64_x6A[VoltageChangeReq]. This field indexes into D18F3x15C as follows:<br>Bits — VID code<br>00b — D18F3x15C[SclkVidLevel0]<br>01b — D18F3x15C[SclkVidLevel1]<br>10b — D18F3x15C[SclkVidLevel2]<br>11b — D18F3x15C[SclkVidLevel3] |
| 2 | **VoltageChangeReq**. Read-write. Software toggles this field to make VDDNB voltage requests. |
| 1 | **VoltageChangeEn**. Read-write. BIOS: 1. Specifies whether changes to D0F0x64_x6A[VoltageChangeReq] causes voltage change requests. 1=Requests occur. 0=Requests do not occur. |
| 0 | **VoltageForceEn**. Read-write. If D0F0x64_x6A[VoltageChangeEn]==1, this field specifies whether changes to D0F0x64_x6A[VoltageChangeReq] cause forced voltage changes. 1=Voltage changes are forced. 0=Voltage changes are not forced. |

### D0F0x64_x6B Voltage Status

Cold reset: 0000_0006h. See 2.5.1.5.2 [Software-Initiated Voltage Transitions].

| Bits | Description |
|---|---|
| 31:3 | Reserved. |
| 2:1 | **CurrentVoltageLevel**. Read-only. Specifies the current voltage level requested by D0F0x64_x6A. See D0F0x64_x6A[VoltageLevel]. To determine the current voltage level, software must poll on this field until two consecutive reads return the same value. |
| 0 | **VoltageChangeAck**. Read-only. Specifies whether the voltage change requested by D0F0x64_x6A[VoltageChangeReq] is complete. |

### D0F0x78 Scratch Register

Cold reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:0 | **Scratch**. Read-write. This register does not control any hardware. |

### D0F0x7C IOC Configuration Control Register

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:1 | Reserved. |
| 0 | **ForceIntGFXDisable: internal graphics disable**. Read-write. Setting this bit disables bridge device 1 on bus 0 and all devices behind this bridge. |

### D0F0x84 Link Arbitration

Reset: 0300_0000h.

| Bits | Description |
|---|---|
| 31:10 | Reserved. |
| 9 | **PmeTurnOff: PME_Turn_Off message trigger**. Read-write. 1=Trigger a PME_Turn_Off message to all downstream devices if PmeMode=1. |
| 8 | **PmeMode: PME message mode**. Read-write. BIOS: 0. 1=PME_Turn_Off message is triggered by writing PmeTurnOff. 0=PME_Turn_Off message is triggered by a message from the FCH. |
| 7:5 | Reserved. |
| 4 | **Ev6Mode: EV6 mode**. Read-write. BIOS: 1. 1=The links decode the memory range from 640K to 1M. |
| 3:0 | Reserved. |

### D0F0x90 Northbridge Top of Memory

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:23 | **TopOfDram**. Read-write. BIOS: MSRC001_001A[TOM[31:23]]. Specifies the address that divides between MMIO and DRAM. From TopOfDram to 4G is MMIO; below TopOfDram is DRAM. See 2.4.3 [Access Type Determination]. |
| 22:0 | Reserved. |

### D0F0x94 Northbridge ORB Configuration Offset

Reset: 0000_0000h.
The index/data pair D0F0x94 and D0F0x98 are used to access D0F0x98_x[FF:00]. To read or write to one of these register, the address is written first into the address register D0F0x94 and then the data are read or written by read or write the data register D0F0x98.

| Bits | Description |
|---|---|
| 31:9 | Reserved. |
| 8 | **OrbIndWrEn: ORB index write enable**. Read-write. 1=Writes to D0F0x98 are enabled. |
| 7 | Reserved. |
| 6:0 | **OrbIndAddr: ORB index register address**. Read-write. |

### D0F0x98 Northbridge ORB Configuration Data Port

See D0F0x94.

### D0F0x98_x06 ORB Downstream Control 0

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:27 | Reserved. |
| 26 | **UmiNpMemWrEn**. Read-write. BIOS: See 2.11.4.1. 1=NP protocol over UMI for memory-mapped writes targeting LPC enabled. This bit may be set to avoid a deadlock condition. |
| 25:0 | Reserved. |

### D0F0x98_x07 ORB Upstream Arbitration Control 0

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:16 | Reserved. |
| 15 | **DropZeroMaskWrEn**. Read-write. BIOS: 1. 1=Drop byte write request that have all bytes masked. 0=Forward byte write request that have all bytes masked. |
| 14 | **MSIHTIntConversionEn**. Read-write. BIOS: 1. 1=MSI to HT interrupt conversion enabled. |
| 13:1 | Reserved. |
| 0 | **IocBwOptEn**. Read-write. BIOS: 1b. 1=Enable optimization of byte writes by detecting consecutive DW masks and translating the request to DW writes. |

### D0F0x98_x08 ORB Upstream Arbitration Control 1

Reset: 0008_0008h. BIOS: 0001_0008h.

This register specifies the weights of the weighted round-robin arbiter in stage 1 of the upstream arbitration for non-posted reads.

| Bits | Description |
|---|---|
| 31:24 | Reserved. |
| 23:16 | **NpWrrLenC**. Read-write. This field defines the maximum number of non-posted read requests from the GPU that are serviced before the arbiter switches to the next client. |

| | |
|---|---|
| 15:8 | Reserved. |
| 7:0 | **NpWrrLenA**. Read-write. This field defines the maximum number of non-posted read requests from IOC that are serviced before the arbiter switches to the next client. |

**D0F0x98_x09 ORB Upstream Arbitration Control 2**

Reset: 0800_0008h. BIOS: 0100_0008h.

This register specifies the weights of the weighted round-robin arbiter in stage 1 of the upstream arbitration for posted writes.

| Bits | Description |
|---|---|
| 31:24 | **PWrrLenD**. Read-write. This field defines the maximum number of posted write requests from the GPU that are serviced before the arbiter switches to the next client. |
| 23:8 | Reserved. |
| 7:0 | **PWrrLenA**. Read-write. This field defines the maximum number of posted write requests from the IOC that are serviced before the arbiter switches to the next client. |

**D0F0x98_x0C ORB Upstream Arbitration Control 5**

Reset: 0000_0808h.

This register specifies the weights of the weighted round-robin arbiter in stage 2 of the upstream arbitration.

| Bits | Description |
|---|---|
| 31 | Reserved. |
| 30 | **StrictSelWinnerEn**. Read-write. BIOS: 1. 1=Select arbitration winner when TX is idle and the FIFO is not full. 0=Select arbitration winner when TX is idle. |
| 29:16 | Reserved. |
| 15:8 | **GcmWrrLenB**. Read-write. BIOS: 08h. This field defines the maximum number of posted write requests from stage 1 that are getting serviced in the round-robin before the stage 2 arbiter switches to the next client. |
| 7:0 | **GcmWrrLenA**. Read-write. BIOS: 08h. This field defines the maximum number of non-posted read requests from stage 1 that are getting serviced in the round-robin before the stage 2 arbiter switches to the next client. |

**D0F0x98_x0E ORB MSI Interrupt Remap**

Reset: 0000_FF0Ch.

| Bits | Description |
|---|---|
| 31:24 | Reserved. |
| 23:16 | **MsiHtRsvIntVector**. Read-write. BIOS: 00h. This field defines the interrupt vector used when an MSI interrupt is received that has a reserved delivery mode field if MsiHtRsvIntRemapEn==1. |

| | |
|---|---|
| 15:8 | **MsiHtRsvIntDestination**. Read-write. BIOS: FFh. This field defines the interrupt destination used when an MSI interrupt is received that has a reserved delivery mode field when MsiHtRsvIntRemapEn==1. |
| 7 | Reserved. |
| 6 | **MsiHtRsvIntDM**. Read-write. BIOS: 0. Defines the interrupt destination mode when an MSI interrupt is received that has a reserved delivery mode field if MsiHtRsvIntRemapEn==1. |
| 5 | **MsiHtRsvIntRqEoi**. Read-write. BIOS: 0. Specifies the REQEIOI state when an MSI interrupt is received that has a reserved delivery mode field if MsiHtRsvIntRemapEn==1. |
| 4:2 | **MsiHtRsvIntMt**. Read-write. BIOS: 011b. Specifies the message type used when an MSI interrupt is received that has a reserved delivery mode field if MsiHtRsvIntRemapEn==1. |
| 1 | Reserved. |
| 0 | **MsiHtRsvIntRemapEn**. Read-write. BIOS: 1. 1=Remapping of MSI interrupts with reserved delivery mode to the interrupt programmed into this register enabled. |

### D0F0x98_x1E ORB Receive Control 0

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:2 | Reserved. |
| 1 | **HiPriEn**. Read-write. BIOS: 0. 1=High priority channel enabled. See Table 8. |
| 0 | Reserved. |

### D0F0x98_x28 ORB Transmit Control 0

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:2 | Reserved. |
| 1 | **ForceCoherentIntr**. Read-write. BIOS: 1. 1=Interrupt request are forced to have coherent bit set. |
| 0 | **SmuPmInterfaceEn**. Read-write. BIOS: 1. 1=SMU to ORB power management interface enabled. |

### D0F0x98_x2C ORB Clock Control

Reset: 000F_0000h.

| Bits | Description |
|---|---|
| 31:16 | **WakeHysteresis**. Read-write. BIOS: 64h. Specifies the amount of time hardware waits after ORB becomes idle before deasserting the wake signal to the NB. Wait time = WakeHysteresis * 50ns. Values less than 64h may result in undefined behavior. |
| 15:2 | Reserved. |
| 1 | **DynWakeEn**. Read-write. BIOS: 1. 1=Enable dynamic toggling of the wake signal between ORB and NB. 0=Disable dynamic toggling of the wake signal. |
| 0 | Reserved. |

### D0F0x98_x3A ORB Source Tag Translation Control 2

Reset: 0000_0000h.

Clumping allows device 2 port to have up to 64 outstanding non-posted requests when device 3 is not enabled.

| Bits | Description |
|---|---|
| 31:4 | Reserved. |
| 3 | **ClumpingEn**. Read-write. IF (D0F0x64_x0C[Dev3BridgeDis]==1 && D0F0x64_x0C[Dev2BridgeDis]==0) THEN BIOS: 1. ELSE BIOS: 0. ENDIF. 1=Enable clumping of internal unit IDs 2 and 3. |
| 2:0 | Reserved. |

### D0F0x98_x4[A:9] ORB LCLK Clock Control 1-0

Reset: 7F3F_8100h. See 2.5.5.3 [GPU and Root Complex Clock Gating].

| Bits | Description |
|---|---|
| 31 | Reserved. |
| 30 | **SoftOverrideClk0**. Read-write. BIOS: 0. See: SoftOverrideClk6. |
| 29 | **SoftOverrideClk1**. Read-write. BIOS: 0. See: SoftOverrideClk6. |
| 28 | **SoftOverrideClk2**. Read-write. BIOS: 0. See: SoftOverrideClk6. |
| 27 | **SoftOverrideClk3**. Read-write. BIOS: 0. See: SoftOverrideClk6. |
| 26 | **SoftOverrideClk4**. Read-write. BIOS: 0. See: SoftOverrideClk6. |
| 25 | **SoftOverrideClk5**. Read-write. BIOS: 0. See: SoftOverrideClk6. |
| 24 | **SoftOverrideClk6**. Read-write. BIOS: 0. 1=Clock gating disabled. 0=Clock gating enabled. |
| 23:12 | Reserved. |
| 11:4 | **OffHysteresis**. Read-write. Specifies the number of LCLK cycles hardware waits after logic goes idle before gating off the clocks specified by SoftOverrideClk[6:0]. |
| 3:0 | Reserved. |

### D0F0x98_x4B ORB SCLK Clock Control

Reset: 4020_0100h. See 2.5.5.3 [GPU and Root Complex Clock Gating].

| Bits | Description |
|---|---|
| 31 | Reserved. |
| 30 | **SoftOverrideClk**. Read-write. BIOS: 0. 1=Clock turn off disabled. 0=Clock turn off enabled. |
| 29:12 | Reserved. |
| 11:4 | **OffHysteresis**. Read-write. Specifies the number of SCLK cycles hardware waits after logic goes idle before gating off the clocks specified by SoftOverrideClk. |
| 3:0 | Reserved. |

### D0F0xE0 Link Index Address

Reset: 0130_8001h.

D0F0xE0 and D0F0xE4 are used to access D0F0xE4_x[FFFF_FFFF:0000_0000]. To read or write to one of these register, the address is written first into the address register D0F0xE0 and then the data is read from or written to the data register D0F0xE4.

The phy index registers (D0F0xE4_x[2xxx_xxxx]) mapping to a specific phy, pin or pin group is shown in a table in the register definition. For example, to perform a read or write operation to configure Gfx phy 0 (P_GFX_[T,R]X[P,N][7:0] pin group) compensation, software should program D0F0xE0[31:0]=0121_0000h. Accessing any register number that is not listed in the mapping table may result in undefined behavior.

Some phy registers support broadcast write operations to groups of 4 or 8 lanes. For example, to perform broadcast write operation to configure Gfx Link[3:0] (P_GFX_RX[P,N][3:0] lanes) receiver phase loop filter, software should program D0F0xE0[31:0]=0221_5602h.

| Bits | Description |
|---|---|
| 31:24 | **BlockSelect: block select**. Read-write. This field is used to select the specific register block to access. The encodings supported depends on the FrameType selected.<br>FrameType — Encoding<br>01h — 1=GPP link core, 2=Gfx link core<br>1xh — 1=Phy interface 0, 2=Phy interface 1 (FrameType 11h only)<br>2xh — 1=Phy 0, 2=Phy 1 (FrameType 21h only)<br>3xh — 1=Wrapper |
| 23:16 | **FrameType: frame type**. Read-write. This field is used to select the type of register block to access.<br>Bits — Destination<br>01h — Link core registers<br>1Nh — Phy interface block registers.<br>2Nh — Phy registers.<br>3Nh — Wrapper registers.<br><br>N — Register Block<br>0h — GPP PCIe Links<br>1h — Gfx PCIe links<br>2h — DDI |
| 15:0 | **PcieIndxAddr: index address**. Read-write. |

### D0F0xE4 Link Index Data

See D0F0xE0.

### 3.3.1 IO Link Registers

**D0F0xE4_x0[2:1]01_0002 IO Link Hardware Debug**

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:1 | Reserved. |
| 0 | **HwDebug[0]: ignore DLLPs in L1**. Read-write. BIOS: 1. 1=DLLPs are ignored in L1 so the TXCLK can be turned off. |

**D0F0xE4_x0[2:1]01_0010 IO Link Control 1**

Reset: 8063_0800h.

| Bits | Description |
|---|---|
| 31:13 | Reserved. |
| 12:10 | **RxSbAdjPayloadSize: slave request payload size**. Read-write. BIOS: 100b. Payload size for requests forwarded to the root complex..<br>Bits Definition Bits Definition<br>00xb Reserved. 100b 64 bytes<br>010b 16 bytes 101b Reserved.<br>011b 32 bytes. 11xb Reserved. |
| 9 | **UmiNpMemWrite: memory write mapping enable**. Read-write. IF (REG == D0F0xE4_x0101_0010) THEN BIOS: 1. ELSE BIOS:0. ENDIF. 1=Internal non-posted memory writes are transferred to UMI. |
| 8:1 | Reserved. |
| 0 | **HwInitWrLock: hardware init write lock**. Read-write. 1=Lock HWInit registers. 0=Unlock HWInit registers. |

**D0F0xE4_x0[2:1]01_0011 IO Link Config Control**

Reset: 0000_0007h.

| Bits | Description |
|---|---|
| 31:4 | Reserved. |
| 3:0 | **DynClkLatency: dynamic clock latency**. Read-write. BIOS: See 2.11.4.2.2. Specifies the number of clock cycles after logic goes idle before clocks are gated off. See 2.5.5.3 [GPU and Root Complex Clock Gating]. |

**D0F0xE4_x0[2:1]01_001C IO Link Control 2**

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:11 | Reserved. |

| | |
|---|---|
| 10:6 | **TxArbMstLimit: transmitter arbitration master limit**. Read-write. BIOS: 4h. Defines together with TxArbSlvLimit a round robin arbitration pattern for downstream accesses. TxArbMstLimit defines the weight for downstream CPU requests and TxArbSlvLimit for the downstream read responses. |
| 5:1 | **TxArbSlvLimit: transmitter arbitration slave limit**. Read-write. BIOS: 4h. See TxArbMstLimit for details |
| 0 | **TxArbRoundRobinEn: transmitter round robin arbitration enabled**. Read-write. BIOS: 1. 1=Enable transmitter round robin arbitration. 0=Disable transmitter round robin arbitration. |

**D0F0xE4_x0[2:1]01_0020 IO Link Chip Interface Control**

Reset: 0000_0050h.

| Bits | Description |
|---|---|
| 31:10 | Reserved. |
| 9 | **CiRcOrderingDis: chip interface root complex ordering disable**. Read-write. 0=Root complex ordering logic is enabled. 1=Root Complex ordering logic is disabled. |
| 8:0 | Reserved. |

**D0F0xE4_x0[2:1]01_0040 IO Link Phy Control**

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:16 | Reserved. |
| 15:14 | **PElecIdleMode: electrical idle mode for physical layer**. Read-write. BIOS: 10b. Defines which electrical idle signal is used, either inferred by the link controller or from the phy.<br>Bits — Definition<br>00b — Gen1 - entry:phy, exit:phy; Gen2 - entry:infer, exit:phy<br>01b — Gen1 - entry:infer, exit:phy; Gen2 - entry:infer, exit:phy<br>10b — Gen1 - entry:phy, exit:phy; Gen2 - entry:PHY, exit:phy<br>11b — Reserved |
| 13:0 | Reserved. |

**D0F0xE4_x0[2:1]01_00B0 IO Link Strap Link Strap Control**

Reset: 0000_0001h.

| Bits | Description |
|---|---|
| 31:6 | Reserved. |
| 5 | **StrapF0AerEn**. Read-write. BIOS: 0. 1=AER support enabled. 0=AER support disabled. |
| 4:3 | Reserved. |
| 2 | **StrapF0MsiEn**. Read-write. BIOS: 1. 1=MSI enabled. 0=MSI disabled. |
| 1:0 | Reserved. |

**D0F0xE4_x0[2:1]01_00C0 IO Link Strap Miscellaneous**

| Bits | Description |
|---|---|
| 31:30 | Reserved. |
| 29 | **StrapMstAdr64En**. Read-write. Reset: 0. |
| 28 | **StrapReverseAll**. Read-write. Reset: 0. |
| 27:0 | Reserved. |

**D0F0xE4_x0[2:1]01_00C1 IO Link Strap Miscellaneous**

| Bits | Description |
|---|---|
| 31:2 | Reserved. |
| 1 | **StrapGen2Compliance**. Read-write. Reset: 1. |
| 0 | **StrapLinkBwNotificationCapEn**. Read-write. Reset: 0. |

### 3.3.2 PIF Registers

**D0F0xE4_x0[2:1]1[2:0]_0010 PIF Control**

Reset: 018A_0059h.

**Table 53: Index addresses for** D0F0xE4_x0[2:1]1[2:0]_0010

| D0F0xE0[31:16] | D0F0xE0[15:0] |
|---|---|
| | 0010h |
| 0110h | GPPFCH PIF |
| 0111h | Gfx PIF 0 |
| 0211h | Gfx PIF 1 |
| 0112h | DDI PIF |

| Bits | Description |
|---|---|
| 31:23 | Reserved. |
| 22:20 | **EiCycleOffTime: electrical idle detect cycle mode off time**. Read-write.<br>Bits / Definition / Bits / Definition<br>000b 2us 100b 10us<br>001b 4us 101b Reserved<br>010b 6us 110b Reserved<br>011b 8us 111b Reserved |
| 19:17 | **Ls2ExitTime: LS2 exit time**. Read-write.<br>Bits / Definition / Bits / Definition<br>000b 10us 100b 625ns<br>001b 5us 101b 0s<br>010b 2.5us 110b Reserved<br>011b 1.25us 111b Reserved |

| | |
|---|---|
| 16:8 | Reserved. |
| 7 | **RxDetectTxPwrMode: receiver detection transmitter power mode**. Read-write. 1=Transmitter is powered on. |
| 6 | **RxDetectFifoResetMode: receiver detect FIFO reset mode**. Read-write. BIOS: 0. 1=The transmit FIFO is reset after receiver detection. 0=The transmit FIFO is not reset after receiver detection. |
| 5 | Reserved. |
| 4 | **EiDetCycleMode: electrical idle detect mode**. Read-write. 1=Electrical idle cycle detection mode is enabled in L1. 0=Electrical idle detection is always enabled in L1. |
| 3:0 | Reserved. |

### D0F0xE4_x0[2:1]1[2:0]_0011 PIF Pairing

Reset: 0200_0000h.

**Table 54: Index addresses for** D0F0xE4_x0[2:1]1[2:0]_0011

| D0F0xE0[31:16] | D0F0xE0[15:0] |
|---|---|
| | 0011h |
| 0110h | GPPFCH PIF |
| 0111h | Gfx PIF 0 |
| 0211h | Gfx PIF 1 |
| 0112h | DDI PIF |

| Bits | Description |
|---|---|
| 31:26 | Reserved. |
| 25 | **MultiPif: x16 link**. Read-write. 1=Lanes 7:0 are paired with a second PIF to create a x16 link. |
| 24:17 | Reserved. |
| 16 | **X8Lane70: x8 link lanes 7:0**. Read-write. 1=Lanes 7:0 are paired to create a x8 link. |
| 15:13 | Reserved. |
| 12 | **X4Lane52: x4 link lanes 5:2**. Read-write. 1=Lanes 5:2 are paired to create a x4 link. |
| 11:10 | Reserved. |
| 9 | **X4Lane74: x4 link lanes 7:4**. Read-write. 1=Lanes 7:4 are paired to create a x4 link. |
| 8 | **X4Lane30: x4 link lanes 3:0**. Read-write. 1=Lanes 3:0 are paired to create a x4 link. |
| 7:4 | Reserved. |
| 3 | **X2Lane76: x2 link lanes 7:6**. Read-write. 1=Lanes 7:6 are paired to create a x2 link. |
| 2 | **X2Lane54: x2 link lanes 5:4**. Read-write. 1=Lanes 5:4 are paired to create a x2 link. |
| 1 | **X2Lane32: x2 link lanes 3:2**. Read-write. 1=Lanes 3:2 are paired to create a x2 link. |
| 0 | **X2Lane10: x2 link lanes 1:0**. Read-write. 1=Lanes 1:0 are paired to create a x2 link. |

### D0F0xE4_x0[2:1]1[2:0]_001[3:2] PIF Power Down Control [1:0]

Reset: 0001_0022h.

**Table 55: Index addresses for** D0F0xE4_x0[2:1]1[2:0]_001[3:2]

| D0F0xE0[31:16] | D0F0xE0[15:0] | |
|---|---|---|
| | 0013h | 0012h |
| 0110h | GPPFCH PIF Lanes 7-4 | GPPFCH PIF Lanes 3-0 |
| 0111h | Gfx PIF 0 Lanes 7-4 | Gfx PIF 0 Lanes 3-0 |
| 0211h | Gfx PIF 1 Lanes 7-4 | Gfx PIF 1 Lanes 3-0 |
| 0112h | DDI PIF Lanes 7-4 | DDI PIF Lanes 3-0 |

| Bits | Description |
|---|---|
| 31:29 | **PllPwrOverrideVal: PLL power state override value**. Read-write. See TxPowerStateInTxs2. |
| 28 | **PllPwrOverrideEn: PLL power state override enable**. Read-write. 1=PLL forced to the power state specified by PllPwrOverrideVal. |
| 27 | Reserved. |
| 26:24 | **PllRampUpTime: PLL ramp time**. Read-write.<br>Bits: 000b = 5us; 001b = 10us; 010b = 300us; 011b = 500us; 100b = 0s; 101b = 0s; 110b = Reserved; 111b = Reserved |
| 23:17 | Reserved. |
| 16 | **Tx2p5clkClockGatingEn**. Read-write. 1=The 2.5x TxClk is gated if the lane is idle 0=The 2.5x TxClk is never gated. |
| 15:13 | Reserved. |
| 12:10 | **PllPowerStateInOff: PLL off power state**. Read-write. See: TxPowerStateInTxs2. All links associated with the PLL must be in the off state to transition the PLL to this state. |
| 9:7 | **PllPowerStateInTxs2: PLL L1 power state**. Read-write. See: TxPowerStateInTxs2. All links associated with the PLL must be in L1 to transition the PLL to this state. |
| 6:4 | **RxPowerStateInRxs2: receiver L1 power state**. Read-write. See: TxPowerStateInTxs2. |
| 3 | **ForceRxEnInL0s: force receiver enable in L0s**. Read-write. 1=The phy CDR is always enabled in L0s. |
| 2:0 | **TxPowerStateInTxs2: transmitter L1 power state**. Read-write.<br>Bits: 000b = L0; 001b = LS1; 010b = LS2; 011b = Reserved; 100b = Reserved; 101b = Reserved; 110b = Reserved; 111b = Off |

### D0F0xE4_x0[2:1]1[2:0]_0015 PIF Transmitter Status

Reset: 0000_0000h.

**Table 56: Index addresses for** D0F0xE4_x0[2:1]1[2:0]_0015

| D0F0xE0[31:16] | D0F0xE0[15:0] |
|---|---|
| | 0015h |
| 0110h | GPPFCH PIF |
| 0111h | Gfx PIF 0 |
| 0211h | Gfx PIF 1 |
| 0112h | DDI PIF |

| Bits | Description |
|---|---|
| 31:8 | Reserved. |
| 7 | **TxPhyStatus07: lane 7 TxPhyStatus**. Read-only. See: TxPhyStatus00. |
| 6 | **TxPhyStatus06: lane 7 TxPhyStatus**. Read-only. See: TxPhyStatus00. |
| 5 | **TxPhyStatus05: lane 5 TxPhyStatus**. Read-only. See: TxPhyStatus00. |
| 4 | **TxPhyStatus04: lane 4 TxPhyStatus**. Read-only. See: TxPhyStatus00. |
| 3 | **TxPhyStatus03: lane 3 TxPhyStatus**. Read-only. See: TxPhyStatus00. |
| 2 | **TxPhyStatus02: lane 2 TxPhyStatus**. Read-only. See: TxPhyStatus00. |
| 1 | **TxPhyStatus01: lane 1 TxPhyStatus**. Read-only. See: TxPhyStatus00. |
| 0 | **TxPhyStatus00: lane 0 TxPhyStatus**. Read-only. Returns the state of the TxPhyStatus signal From the PIF to the BIF. |

### 3.3.3 Phy Registers

There are three categories of phy registers: 3.3.3.1 [Global Phy Control Registers], 3.3.3.2 [Phy Receiver Lane Control Registers] and 3.3.3.3 [Phy Transmitter Lane Control Registers].

#### 3.3.3.1 Global Phy Control Registers

Each global phy control register may have one instance per phy or two instances per phy (one per nibble). When a global register is implemented per phy the mapping to signal pins is shown in Table 57. When a global register is implemented per nibble the mapping to pins is shown in Table 57.

**Table 57: Per phy register addresses to pin mappings**

| D0F0xE0[31:0] | Pin Names |
|---|---|
| 0120h_[2:0]xxxh | GPP Links: P_GPP_[T,R]X[P,N][3:0] & FCH ports: P_UMI_[T,R]X[P,N][3:0] |
| 0121h_[2:0]xxxh | Gfx Links[7:0]: P_GFX_[T,R]X[P,N][7:0] |
| 0221h_[2:0]xxxh | Gfx Links[15:8]: P_GFX_[T,R]X[P,N][15:8] |
| 0122h_[2:0]xxxh | DDI 1: DP1_TX[P,N][3:0] & DDI 0: DP0_TX[P,N][3:0] |

**Table 58: Per nibble register addresses to pin mappings**

| D0F0xE0[31:12] | Pin Names | |
|---|---|---|
| | Address N+1 | Address N |
| 0120h_[2:0]xxxh | GPP ports: P_GPP_[T,R]X[P,N][3:0] | FCH ports: P_UMI_[T,R]X[P,N][3:0] |

**Table 58: Per nibble register addresses to pin mappings**

| | | |
|---|---|---|
| 0121h_[2:0]xxxh | Graphics port lower: P_GFX_[T,R]X[P,N][7:4] | Graphics port lower: P_GFX_[T,R]X[P,N][3:0] |
| 0221h_[2:0]xxxh | Graphics port upper: P_GFX_[T,R]X[P,N][15:12] | Graphics port upper: P_GFX_[T,R]X[P,N][11:8] |
| 0122h_[2:0]xxxh | DDI 0: DP0_TX[P,N][3:0] | DDI 1: DP1_TX[P,N][3:0] |

### D0F0xE4_x0[2:1]2[2:0]_0000 Phy Compensation Control and Calibration Control I

This register provides general control of various circuits that perform auto-calibration.

| Bits | Description |
|---|---|
| 31:28 | Reserved. |
| 27:23 | **RttRawCal: receiver termination resistance (Rtt) raw calibration value**. Read-only. Reset: 0. This field provides the raw Rtt calibration value as determined by the compensation circuit. |
| 22:18 | **RonRawCal: transmitter resistance (Ron) raw calibration value**. Read-only. Reset: 0. This field provides the raw Ron calibration value as determined by the compensation circuit. |
| 17:0 | Reserved. |

### D0F0xE4_x0[2:1]2[2:0]_000[2:1] Phy Impedance Control

Updates to these registers that result in a change to impedance may not take effect in the phy for up to 2 micro-seconds after the update to this register completes.

| Bits | Description |
|---|---|
| 31:29 | **RttCtl: receiver termination resistance (Rtt) control**. Read-write. Reset: 0. This field specifies how the receiver termination resistance value is calculated. All values between 00h and 1Fh are valid.<br>Bits — Definition<br>000b — Rtt is as determined by the compensation circuit, D0F0xE4_x0[2:1]2[2:0]_0000[RttRawCal].<br>001b — Rtt is as specified by RttIndex - 3.<br>010b — Rtt is as specified by the difference: RttRawCal - RttIndex. If this results in a value that is less than 00h, then 00h is used.<br>011b — Rtt is as specified by the sum: RttRawCal + RttIndex. If this results in a value that is greater than 1Fh, then 1Fh is used.<br>100b — Enable only one tap of the Rtt resistor, as specified by RttIndex, and disable the base resistor that is normally always enabled. This is intended for testing purposes only.<br>111b-101b — Reserved.<br><br>For all modes (except 100b), higher values reduce the resistance of Rtt and lower values increase the resistance of Rtt.<br>If RttCtl is programmed to either 011b or 100b, the value of RttRawCal + RttIndex must be less than or equal to 24. |
| 28:21 | Reserved |
| 20:16 | **RttIndex: receiver termination resistance (Rtt) index**. Read-write. Reset: 0. See RttCtl. |

| | |
|---|---|
| 15:13 | **RonCtl: transmitter resistance (Ron) control**. Read-write. Reset: 0. This field specifies how the transmitter resistance value is calculated.<br>Bits — Definition<br>000b — Ron is as determined by the compensation circuit, D0F0xE4_x0[2:1]2[2:0]_0000[RonRawCal].<br>001b — Ron is as specified by the RonIndex field.<br>010b — Ron is as specified by the difference: RonRawCal - RonIndex. If this results in a value that is less than 00h, then 00h is used.<br>011b — Ron is as specified by the sum: RonRawCal + RonIndex. If this results in a value that is greater than 1Fh, then 1Fh is used.<br>100b — Enable only one tap of the Ron resistor, as specified by RonIndex, and disable the base resistor that is normally always enabled. This is intended for testing purposes only.<br>111b-101b — Reserved.<br><br>For all modes (except 100b), higher values reduce the resistance of Ron and lower values increase the resistance of Ron.<br>If RonCtl is programmed to either 011b or 100b, the value of RonRawCal + RonIndex must be less than or equal to 23. |
| 12:5 | Reserved. |
| 4:0 | **RonIndex: transmitter resistance (Ron) index**. Read-write. Reset: 0. See RonCtl. |

### D0F0xE4_x0[2:1]2[2:0]_000[A:9] Phy Clock Tree Control

BIOS: See 2.11.4.2.2 [Link Configuration and Core Initialization] and Table 38 [Supported DDI Modes].

| Bits | Description |
|---|---|
| 31 | **PCIePllSel**. Read-write. Reset: 1. 1=Selects PCIe clock. 0=Selects the display clock. |
| 30:29 | Reserved. |
| 28 | **CascadedPllSel**. Read-write. Reset: 0. 1=Selects cascaded PLL clocks. 0=Selects the external display PLL clocks. |
| 27:26 | Reserved. |
| 25 | **DisplayStream**. Read-write. Reset: 0. 1=Selects display PLL1 and cascaded PLL clock 1. 0=Selects display PLL0 and cascaded PLL clock 0. |
| 24 | **ClkOff**. Read-write. Reset: 0. 1=Turns off clock to 4 lane wide sub-link within phy. 0=Enables clock to 4 lane wide sublink within phy. |
| 23:0 | Reserved. |

### D0F0xE4_x0[2:1]2[2:0]_000[C:B] Phy Serial Bus Packet Control

This register provides control to enable or disable various fields contained in the phy serial bus primary control packet, the margining packet and the miscellaneous control packet.

| Bits | Description |
|---|---|
| 31:16 | Reserved. |
| 15 | **PllCmpPktSbiEn**. Read-write. Reset: 1. 1=Enables the serial bus PLL component packet used for controlling PLL features such as mode of operation and power states. |

| | |
|---|---|
| 14 | **MargPktSbiEn**. Read-write. Reset: 1. IF (REG==D0F0xE4_x0122_000[C:B]) THEN BIOS: 0. ENDIF. 1=Enables the serial bus margining update packet used for controlling PCIe transmit margining test. |
| 13:9 | Reserved. |
| 8 | **EiDetSbiEn**. Read-write. Reset: 1. 1=Enables the electrical idle detector control field in the primary control packet. |
| 7 | **IncoherentClkSbiEn**. Read-write. Reset: 1. 1=Enables the incoherent clock control field in the primary control packet. |
| 6 | **SkipBitSbiEn**. Read-write. Reset: 1. 1=Enables the skip bit control field in the primary control packet. |
| 5 | **OffsetCancelSbiEn**. Read-write. Reset: 1. 1=Enables the offset cancellation control field in the primary control packet. |
| 4 | **DllLockSbiEn**. Read-write. Reset: 1. 1=Enables the DLL lock control field in the primary control packet. |
| 3 | **FreqDivSbiEn**. Read-write. Reset: 1. 1=Enables the frequency divider control field in the primary control packet. |
| 2 | **PcieModeSbiEn**. Read-write. Reset: 1. IF(REG==D0F0xE4_x0122_000[C:B]) THEN BIOS: 0. ENDIF. 1=Enables the phy mode control field in the primary control packet. |
| 1 | **RxPwrSbiEn**. Read-write. Reset: 1. 1=Enables the Rx power state control field in the primary control packet. |
| 0 | **TxPwrSbiEn**. Read-write. Reset: 1. 1=Enables the Tx power state control field in the primary control packet. |

#### D0F0xE4_x0[2:1]2[2:0]_000D Phy Serial Bus Compensation Component Packet Enable

This register provides control to enable or disable the phy serial bus compensation component packet.

| Bits | Description |
|---|---|
| 31:1 | Reserved. |
| 0 | **CmpCmpPktSbiEn**. Read-write. Reset: 1. 1=Enables the serial bus compensation component packet used for controlling compensation circuit features and power states. |

#### D0F0xE4_x0[2:1]2[2:0]_2000 Phy PLL Power State Control

This register provides control of the phy PLL component power state.

| Bits | Description |
|---|---|
| 31:4 | Reserved. |

| | |
|---|---|
| 3 | **PllAutoPwrDownDis**. Read-write. Reset: 0. 1=Disables the automatic power down feature. 0=Enables the automatic power down feature; PLL powers down when it determines that it is unused. |
| 2:0 | **PllPowerDownEn**. Read-write. Reset: 0. This filed specifies the power state of the phy PLL circuit component.<br>Bits — Definition<br>000b — L0 power state; all circuits are enabled.<br>001b — LS1 power state; all circuits are enabled.<br>010b — LS2 power state; PLL is powered down and clock tree is gated off.<br>110b-011b — Reserved.<br>111b — PHYOFF power state; all circuits are disabled to achieve the lowest power consumption. |

### D0F0xE4_x0[2:1]2[2:0]_2002 Phy PLL Control

The newly written values to this register do not take effect until an explicit shadow update event takes place via setting D0F0xE4_x0[2:1]2[2:0]_2008[PllControlUpdate], or until the PLL component exits the LS2 or PHYOFF power state. Any read from this register always returns the current register value, not the value pending until the next update event. See D0F0xE4_x0[2:1]2[2:0]_2008[PllControlUpdate].

| Bits | Description |
|---|---|
| 31 | **IsLc: PLL select**. Read-write. Reset: 0. This bit selects the raw PLL. 1=Selects the LC tank raw PLL. 0=Selects the ring oscillator raw PLL. |
| 30:28 | Reserved. |
| 27 | **RoCalEn: Ring oscillator calibration enable**. Read-write. Reset: 0. BIOS: 1. A 0 to 1 edge transition of this bit triggers a calibration cycle for the ring oscillator VCO. Reset and set this bit again to trigger another calibration cycle if needed. |
| 26:0 | Reserved. |

### D0F0xE4_x0[2:1]2[2:0]_2005 Phy PLL Frequency and Mode Control

This register provides PLL operation mode control and frequency selection. The newly written values to this register do not take effect until an explicit shadow update event takes place via setting D0F0xE4_x0[2:1]2[2:0]_2008[PllControlUpdate], or until the PLL component exits the LS2 or PHYOFF power state. Any read from this register always returns the current register value, not the value pending until the next update event. See D0F0xE4_x0[2:1]2[2:0]_2008[PllControlUpdate].

| Bits | Description |
|---|---|
| 31:15 | Reserved. |
| 14:13 | **PllMode: PLL operation mode**. Read-write. Reset: 0. BIOS: See 2.11.4.2.1 [Clock Configuration].<br>Bits — Definition<br>00b — PCIe mode<br>01b — Reserved<br>10b — IsCascadedClk0 mode<br>11b — IsCascadedClk1 mode |
| 12:11 | Reserved. |
| 10:9 | **PllClkFreqExt**. Read-write. Reset: 10b. See PllClkFreq. |

| | |
|---|---|
| 8:4 | Reserved. |
| 3:0 | **PllClkFreq: PLL frequency select**. Read-write. Reset: 0011b. BIOS: See 2.11.4.2.1 [Clock Configuration]. This field is used together with PllClkFreqExt to specify the PLL frequency.<br>{PllClkFreqExt, PllClkFreq} — Definition<br>100000b — 810MHz (display port)<br>100011b — 2.5GHz (PCIe Gen 2)<br>110000b — 125MHz - 156.25MHz (DVI/HDMI)<br>110001b — 156.25MHz - 206.25MHz (DVI/HDMI)<br>110010b — 206.25MHz - 250MHz (DVI/HDMI)<br>110011b — 250MHz - 312.5MHz (DVI/HDMI)<br>110100b — 312.5MHz - 412.5MHz (DVI/HDMI)<br>110101b — 412.5MHz - 500MHz (DVI/HDMI)<br>110110b — 500MHz - 625MHz (DVI/HDMI)<br>110111b — 625MHz - 825MHz (DVI/HDMI)<br>111000b — 825MHz - 1125MHz (DVI/HDMI)<br>all others — Reserved |

**D0F0xE4_x0[2:1]2[2:0]_2008 Phy PLL Update Control**

| Bits | Description |
|---|---|
| 31:30 | Reserved. |
| 29 | **VdDetectEn: PLL output clock gating enable**. Read-write. Reset: 0. BIOS: 1. 1=Enables clock gating during LS2 exit. The internally regulated power supply to the phy is turned off and then back on during LS2 entry and exit. A power sniffer circuit detects the internally regulated power supply level. PLL output clock distribution can be gated off before the regulated power supply reaches the desired level during power up, this effectively prevents possible reliability issue arising from undesirable drivers contention and crow bar current stress during power up upon LS2 exit. Clock distribution is turned on only when the regulated power reaches the right level. 0=PLL output clock distribution is not gated during power up upon LS2 exit. |
| 28:26 | Reserved. |
| 25:23 | **MeasCycCntVal[2:0]**. Read-write. Reset: 0. This field specifies the time interval over which the LC VCO feedback clock cycles is counted.<br>Bits — Definition<br>000b — 3167 count for 31.5kHz modulation<br>001b — 3023 count for 33.0kHz modulation (max spec)<br>010b — 3325 count for 33.0 kHz modulation (min spec)<br>011b — 6334 count, 2 times default (long count)<br>100b — 12667 count, 4 times default (very long count)<br>101b — 2048, 11bits (shorter count for debug)<br>110b — 1024, 10bits<br>111b — 512, 9bits |
| 22:1 | Reserved. |
| 0 | **PllControlUpdate: PLL control register update**. Read-write. Reset: 0. 0 to 1 transition of this bit triggers the explicit shadow update event for the phy PLL registers that are shadowed. |

### 3.3.3.2 Phy Receiver Lane Control Registers

Each receiver lane has a group of registers for controlling the operation of the lane. The mapping from address

register to receiver lane is shown in Table 59. Multiple receiver lanes may be written at the same time using per nibble and per byte broadcast write addresses. The mapping from broadcast address to receiver lanes is shown in Table 60.

**Table 59: Phy per receiver lane register addresses**

| Pin Group | D0F0xE0[31:16] | D0F0xE0[15:0] | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 438xh | 430xh | 428xh | 420xh | 418xh | 410xh | 408xh | 400xh |
| P_GPP_ | 0120h | RX[P,N]3 | RX[P,N]2 | RX[P,N]1 | RX[P,N]0 | - | - | - | - |
| P_UMI_ | 0120h | - | - | - | - | RX[P,N]3 | RX[P,N]2 | RX[P,N]1 | RX[P,N]0 |
| P_GFX_ | 0121h | RX[P,N]7 | RX[P,N]6 | RX[P,N]5 | RX[P,N]4 | RX[P,N]3 | RX[P,N]2 | RX[P,N]1 | RX[P,N]0 |
| P_GFX_ | 0221h | RX[P,N]15 | RX[P,N]14 | RX[P,N]13 | RX[P,N]12 | RX[P,N]11 | RX[P,N]10 | RX[P,N]9 | RX[P,N]8 |

**Table 60: Phy receiver broadcast register addresses**

| D0F0xE0[31:16] | D0F0xE0[15:0] | | |
|---|---|---|---|
| | 57[1,0]xh | 56[1,0]xh | 50[1,0]xh |
| 0120h | P_GPP_RX[P,N][3:0] | P_UMI_RX[P,N][3:0] | P_GPP_RX[P,N][3:0], P_UMI_RX[P,N][3:0] |
| 0121h | P_GFX_RX[P,N][7:4] | P_GFX_RX[P,N][3:0] | P_GFX_RX[P,N][7:0] |
| 0221h | P_GFX_RX[P,N][15:12] | P_GFX_RX[P,N][11:8] | P_GFX_RX[P,N][15:8] |

### D0F0xE4_x0[2:1]2[1:0]_[5:4][7:6,3:0][8,0]1 Phy Receiver DLL Control and Test 1

These registers provide control of the DLL and duty cycle correction circuit associated with the receive lanes on the phys.

| Bits | Description |
|---|---|
| 31:16 | Reserved. |
| 15 | **ForceDccRecalc: Force DCC code recalculation**. Read-write. Reset: 0. BIOS: 1. A 0 to 1 edge transition of this bit forces the DCC code to be recalculated when the DLL core loop is locked the next time. This could be useful for debug. When transitioning between the RO PLL to the LC PLL this bit should be set. |
| 14:0 | Reserved. |

### D0F0xE4_x0[2:1]2[1:0]_[5:4][7:6,3:0][8,0]2 Phy Receiver Phase Loop Filter Control

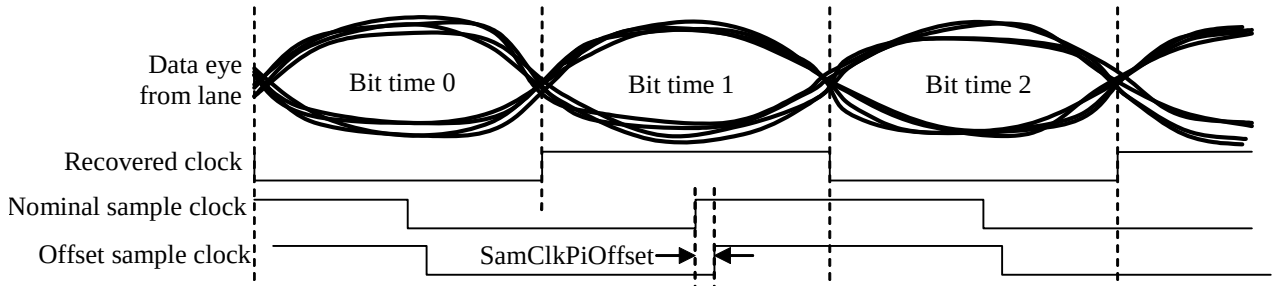


**Figure 13: Phy recovered clock and sample clock**

When the link is in a mode that relies on dynamic phase alignment (automatic sample-clock correction), then the processor generates a recovered clock for each lane based on transitions in the lane. The ideal recovered

clock transitions at exactly the same time as the transitions in the lane. Phase detection logic detects if the recovered clock transitions before or after the lane transition. The digital loop filter (DLF) is logic that adjusts the phase of the recovered clock such that its transitions match the transition time of the lane as much as possible. The DLF counts the number of times the lane transitions before the recovered clock versus after to determine whether the recovered clock phase requires adjustment. The DLF uses an 8-bit counter, called the loop filter counter (LFC) for this purpose. The LFC controls are included in this register. They specify DLF behavior as follows:

- LfcMax is programmed to be greater than LfcMin.
- The LFC is initialized to LfcMin.
- The LFC is updated periodically. The logic keeps a tally of the number of lane transitions occurring before and after the recovered clock transition within each update period.
- To start, if there is a net lane transition occurs after the recovered clock transition within the update period, the LFC is incremented by the net value; on the other hand, if there is a net lane transition occurs before the recovered clock transition, the LFC is decremented. However, if the LFC is ever decremented while it is zero, these rules are reversed (and the LFC is incremented instead). Thus, if there is a phase correction needed, the LFC trends either upward or downward; if it trends downward, it hits zero and then trends upward again.
- If the LFC reaches LfcMax value, then (1) the phase of the recovered clock is adjusted in the appropriate direction, (2) the LFC is set to the LfcMin value.

The LfcMin and LfcMax fields are designed to improve the stability of the recovered clock phase while improving the response time for multiple phase updates in the same direction. For example, if the recovered clock phase needs several adjustments in the same direction, then the LFC increments until it hits LfcMax value and then be set to LfcMin (and trigger a phase adjustment); then it would increment to LfcMax value again to trigger the next phase adjustment. If, however, the next phase adjustment needs to be in the opposite direction, the LFC would decrement to zero, change direction, and then increment up to LfcMax again. In this way, phase adjustments in the same direction occur more quickly than phase adjustments in the opposite direction of the prior phase adjustment.

The nominal sample clock is offset by 90 degrees from the *recovered clock*. An offset can be inserted to move the sample clock from the nominal position, based on SamClkPiOffset and SamClkPiOffsetSign.

| Bits | Description |
|---|---|
| 31:30 | Reserved. |
| 29:22 | **LfcMax: loop filter counter maximum value**. Read-write. Reset: 08h. BIOS: 08h. |
| 21:14 | **LfcMin: loop filter counter minimum value**. Read-write. Reset: 00h. BIOS: 00h. |
| 13:8 | Reserved. |
| 7 | **SamClkPiOffsetEn: sample clock phase interpolator offset enable**. Read-write. Reset: 0. 1=Enable offset insertion around the nominal sample clock position. |
| 6:4 | **SamClkPiOffset: sample clock phase interpolator offset setting**. Read-write. Reset: X. This field specifies the magnitude of the offset of the sample clock from the nominal position. See Figure 13. This field is encoded as follows.<br>• Sample clock phase interpolator offset = (SamClkPiOffset + 1) * step size.<br>• If link speed is >3.6GT/s, the expected typical step size is 2ps with a +/-1ps error.<br>• If link speed is <=3.6GT/s, the expected typical step size is 3ps with a +/-1ps error. |

| | |
|---|---|
| 3 | **SamClkPiOffsetSign: sample clock phase interpolator offset setting sign bit**. Read-write. Reset: X. 0=Sample clock is moved to before the nominal position. 1=Sample clock is moved to after the nominal position. See SamClkPiOffset and Figure 13. |
| 2:0 | Reserved. |

### D0F0xE4_x0[2:1]2[1:0]_[5:4][7:6,3:0][8,0]5 Phy Receiver Timing Margin Test

The built in jitter injection test mode is useful for checking the clock data recovery tracking bandwidth of the receiver. By forcing the sample clock to move from the lock position by a controlled amount and then observing the time it takes to recover, the tracking rate and bandwidth can be estimated. This register provides the control of the test mode.

The jitter injection test mode works as follows.

- The circuit is clocked by a jitter injection clock derived from dividing the link forwarded clock by 2.5; for example, if the link speed is 5.2GT/s and the link forwarded clock frequency is 2.6GHz, the jitter injection clock frequency becomes 1.04GHz.
- There are 2 phases, the on phase and the off phase. It starts with the on phase once the test mode is enabled.
- During the on phase, at every tick of jitter injection clock, the sample clock is moved away from the nominal lock position by 1/96*UI.
- The direction of adjustment is specified by JitterInjDir.
- The on phase adjustment continues for a number of times as specified by JitterInjOnCnt.
- Then the adjustment turns off for a duration specified by {JitterInjOffCnt, JitterInjOnCnt} * jitter injection clock period, this is known as the off phase. During this time, clock data recovery resumes to try to adjust the position of the sample clock back to the center of the data eye.
- The off phase is followed by the on phase again. The process continues to alternate between the on phase and the off phase until the jitter injection test mode is disabled.

In addition, the JitterInjHold bit may be set to inject a hold state at the end of the on phase. This stops clock data recovery from resuming after the on phase, hence holding the sample clock at its last adjusted position until the JitterInjHold bit is cleared. This test mode may be useful for margining the width of the input data eye.

This margining mechanism is not characterized for precision jitter adjustments or measurements.

| Bits | Description |
|---|---|
| 31 | Reserved. |
| 30 | **JitterInjEn: jitter injection enable**. Read-write. Reset: 0. 1=Jitter injection test mode is enabled. |
| 29 | **JitterInjDir: jitter injection direction**. Read-write. Reset: 0.<br>Bit Definition<br>0 Move clock before the nominal lock position.<br>1 Move clock after the nominal lock position. |
| 28:23 | **JitterInjOnCnt: jitter injection on count**. Read-write. Reset: 0. |
| 22:16 | Reserved. |
| 15:10 | **JitterInjOffCnt: jitter injection off count**. Read-write. Reset: 0. The jitter injection off time count is a 12bit code, this field specifies the most significant 6 bits. The least significant 6 bits are the same as JitterInjOnCnt. |
| 9 | **JitterInjHold: jitter injection hold**. Read-write. Reset: 0. 1=Jitter injection hold is enabled. |
| 8:0 | Reserved. |

### D0F0xE4_x0[2:1]2[1:0]_[5:4][7:6,3:0][8,0]6 Phy Receiver DFE and DFR Control

The processor supports decision feedback restore (DFR), a function that enables on-chip AC coupling on the receiver path, to improve the receiver's ability to operate over a longer channel. In this mode, the receiver on the processor must be programmed with the expected peak single-ended DC voltage level over the single-ended DC common mode voltage level, as seen by the receiver, when a static 1 or 0 is driven. For example, without deemphasis at nominal supply voltage of 1.2V, the peak single ended voltage is expected to be 300mV ideally above the single ended DC common mode voltage level. The value is dependent on the deemphasis setting of the transmitter on the other end of the channel.

**Table 61: Recommended DCV settings**

| Far-device deemphasis setting | DCV |
|---|---|
| No deemphasis | 20h |
| -3dB postcursor | 17h |
| -6dB postcursor | 10h |

Decision feedback equalization (DFE) can be enabled to enhance link operation. Once enabled, the receiver uses the logic level of the previous data bit to adjust the voltage threshold of the sampler in the direction that causes the sampler to switch sooner when the data bit transitions to the opposite logic level for the next bit. The control and DFE voltage level are included in this register.

| Bits | Description |
|---|---|
| 31:8 | Reserved. |
| 7 | **DfeEn: DFE enable**. Read-write. Reset: 0. 1=Decision feedback equalization is enabled. |
| 6:5 | **DfeVoltage: DFE offset voltage level**. Read-write. Reset: 0. This field specifies the magnitude of the DFE offset voltage.<br>Bits Definition Bits Definition<br>00b DFE offset voltage=25mV. 10b DFE offset voltage=12.5mV.<br>01b DFE offset voltage=0mV. 11b DFE offset voltage=31.25mV. |
| 4:0 | Reserved. |

### D0F0xE4_x0[2:1]2[1:0]_[5:4][7:6,3:0][8,0]A Phy DLL Test and Control 3

| Bits | Description |
|---|---|
| 31:29 | **Ls2ExitTime: LS2 exit time**. Read-write. Reset: 0. BIOS: 001b. This field selects the internal timer that delays the turn-on of the DLL after exit from LS2 state to L0 state. The added delay allows the forwarded input clock to achieve better stability.<br>Bits Definition<br>000b Delay=10us.<br>001b Delay=5us.<br>010b Delay=2.5us.<br>011b Delay=1.25us.<br>100b Delay=625ns.<br>101b Delay=0s.<br>11xb reserved.<br>The value specified by Ls2ExitTime must be less than the value specified by T0Time, or it can cause undefined behavior. |

| | |
|---|---|
| 28:18 | Reserved. |
| 17 | **DllLockFastModeEn: DLL lock fast mode enable**. Read-write. Reset: 0. 1=Enables DLL lock fast mode. 0=DLL lock operates at standard speed. |
| 16:15 | Reserved. |
| 14:13 | **AnalogWaitTime: analog wait time to turn on DLL**. Read-write. Reset: 0. The turning on of the DLL circuit after cold reset is delayed by a timer specified by this field. The encodings are as follows:<br>Bits Definition Bits Definition<br>00b Delay=1.25us. 10b Delay=2.5us.<br>01b Delay=0.625us. 11b Delay=0.3125us. |
| 12:8 | Reserved. |
| 7 | **BiasDisInLs2: bias disable in LS2 power state**. Read-write. Reset: 0. IF ((REG==D0F0xE4_x0121_4[1:0][8,0]A && D0F0xE4_x0131_8040[OwnPhyA]==0) \|\| (REG==D0F0xE4_x0121_4[3:2][8,0]A && D0F0xE4_x0131_8040[OwnPhyB]==0) \|\| (REG==D0F0xE4_x0221_4[1:0][8,0]A && D0F0xE4_x0131_8040[OwnPhyC]==0) \|\| (REG==D0F0xE4_x0221_4[3:2][8,0]A && D0F0xE4_x0131_8040[OwnPhyD]==0) \|\| REG==D0F0xE4_x0120_4[1:0][8,0]A \|\| REG==D0F0xE4_x0120_4[3:2][8,0]A) THEN BIOS: 1. ENDIF. 1=Enables lower power LS2 state; current consumption is lowered by approximately 2.5mA per receive lane when compared to standard LS2 power mode. Setting this bit increases the amount of T0Time needed to relock the DLL. When this bit is set, Ls2ExitTime must be programmed to select a value that is greater than or equal to AnalogWaitTime. 0=Standard LS2 power mode. |
| 6:5 | Reserved. |
| 4 | **LockDetOnLs2Exit: DLL lock detect on LS2 exit**. Read-write. Reset: 0. This field selects the LS2 to L0 power state transition speed. 1=Fast transition mode selected. 0=Slow transition mode selected. |
| 3:2 | Reserved. |
| 1 | **RxPcieMode: receiver PCIe mode**. Read-write; set-by-hardware. Cold reset: 0. This field indicates the receiver PCIe mode. 0=Gen 1. 1=Gen 2. |
| 0 | **EnCoreLoopFirst: enable DLL core loop first on LS2 exit**. Read-write. Reset: 0. This field selects LS2 to L0 power state transition speed. 1=Fast transition mode selected. 0=Slow transition mode selected. |

#### 3.3.3.3 Phy Transmitter Lane Control Registers

Each transmitter lane has a group of registers for controlling the operation of the lane. The mapping from address register to transmitter lane is shown in Table 62. Multiple transmitter lanes may be written at the same time using per nibble and per byte broadcast write addresses. The mapping from broadcast address to transmitter lanes is shown in Table 63.

**Table 62: Phy per transmitter lane register addresses**

| Pin Group | D0F0xE0[31:16] | D0F0xE0[15:0] | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 638xh | 630xh | 628xh | 620xh | 618xh | 610xh | 608xh | 600xh |
| P_GPP_ | 0120h | TX[P,N]3 | TX[P,N]2 | TX[P,N]1 | TX[P,N]0 | - | - | - | - |
| P_UMI_ | 0120h | - | - | - | - | TX[P,N]3 | TX[P,N]2 | TX[P,N]1 | TX[P,N]0 |
| P_GFX_ | 0121h | TX[P,N]7 | TX[P,N]6 | TX[P,N]5 | TX[P,N]4 | TX[P,N]3 | TX[P,N]2 | TX[P,N]1 | TX[P,N]0 |
| P_GFX_ | 0221h | TX[P,N]15 | TX[P,N]14 | TX[P,N]13 | TX[P,N]12 | TX[P,N]11 | TX[P,N]10 | TX[P,N]9 | TX[P,N]8 |

**Table 62: Phy per transmitter lane register addresses**

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| P_DP0_ | 0122h | TX[P,N]3 | TX[P,N]2 | TX[P,N]1 | TX[P,N]0 | - | - | - | - |
| P_DP1_ | 0122h | - | - | - | - | TX[P,N]3 | TX[P,N]2 | TX[P,N]1 | TX[P,N]0 |

**Table 63: Phy transmitter broadcast register addresses**

| D0F0xE0[31:16] | D0F0xE0[15:0] | | |
|---|---|---|---|
| | 77[1,0]xh | 76[1,0]xh | 70[1,0]xh |
| 0120h | P_GPP_TX[P,N][3:0] | P_UMI_TX[P,N][3:0] | P_GPP_TX[P,N][3:0], P_UMI_TX[P,N][3:0] |
| 0121h | P_GFX_TX[P,N][7:4] | P_GFX_TX[P,N][3:0] | P_GFX_TX[P,N][7:0] |
| 0221h | P_GFX_TX[P,N][15:12] | P_GFX_TX[P,N][11:8] | P_GFX_TX[P,N][15:8] |
| 0122h | DP0_TX[P,N][3:0] | DP1_TX[P,N][3:0] | DP1_TX[P,N][3:0], DP0_TX[P,N][3:0] |

### D0F0xE4_x0[2:1]2[2:0]_[7:6][7:6,3:0][8,0]0 Link Phy Transmit Control

| Bits | Description |
|---|---|
| 31:16 | Reserved. |
| 15 | **DisLoImpIdle: disable low impedance idle**. Read-write. Reset: 0. 1= Disables the low impedance electrical idle feature that requires both the true and complement pins of the transmitter to be pulled to VDDP/2 via low impedance termination in the range of 25 to 50 ohm upon entering electrical idle state. Instead, 5k ohm termination is used. 0=Enables low impedance electrical idle mode. |
| 14:8 | Reserved. |
| 7 | **TxLs23ClkGateEn: LS2/LS3 clock gating enable**. Read-write. Reset: 1. 1= Internal phy clock grids are gated during LS2 or PHY OFF states to save power. |
| 6:0 | Reserved. |

### D0F0xE4_x0[2:1]2[2:0]_[7:6][7:6,3:0][8,0]5 Phy Transmit Link Configuration

**Table 64: Recommended link configuration**

| Link configuration | GangedMo-deEn | IsOwnMstr |
|---|---|---|
| x1 (1 lane per sublink) | 0 | 1 |
| x2 (2 lanes per sublink) | 0 | 1 |
| x4 (4 lanes per sublink) | 0 | 0 |
| x8 | 1 | 0 |

| Bits | Description |
|---|---|
| 31 | **GangedModeEn**. Read-write. Reset: 1. BIOS: Table 64. 1=Enables link ganged mode. 0=Disables link ganged mode. |
| 30 | Reserved. |
| 29 | **IsOwnMstr**. Read-write. Reset: 0. BIOS: Table 64. 1=Enables the lane to self initialize its own read pointer. |
| 28:0 | Reserved. |

**D0F0xE4_x0[2:1]2[2:0]_[7:6][7:6,3:0][8,0]6 Phy Transmit Nominal Deemphasis Control**

This register specifies the deemphasis, or preemphasis settings in the case of display port mode, and voltage margining settings for the transmit drivers.

**Table 65: Recommended preemphasis settings**

| Conditions | | | D0F0xE4_x0[2:1]2[2:0]_[7:6][7:6,3:0][8,0]6 | |
|---|---|---|---|---|
| Link Type | Preemphasis | Peak-to-peak Voltage | DeemphGen1Nom | TxMarginNom |
| Display Port | 0dB | 1.2V | 0 | 0 |
| | | 0.8V | 0 | 42 |
| | | 0.6V | 0 | 64 |
| | | 0.4V | 0 | 85 |
| | 3.5dB | 0.8V | 42 | 0 |
| | | 0.6V | 32 | 32 |
| | | 0.4V | 21 | 64 |
| | 6dB | 0.6V | 64 | 0 |
| | | 0.4V | 42 | 42 |
| | 9.5dB | 0.4V | 85 | 0 |
| PCIe \|\| HDMI | - | 1.2V | 42 | 0 |
| DVI | - | 1.2V | 11 | 0 |

| Bits | Description |
|---|---|
| 31:16 | Reserved. |
| 15:8 | **DeemphGen1Nom**. Read-write. Reset: 42. BIOS: Table 65.This field specifies the post cursor deemphasis setting. Value must be less than or equal to 104.. |
| 7:0 | **TxMarginNom**. Read-write. Reset: 0. BIOS: Table 65.This field specifies the voltage margining setting of the transmit driver. Value must be less than or equal to 104. |

### 3.3.4 Wrapper Registers

**D0F0xE4_x013[1:0]_0080 Link Configuration**

| Bits | Description |
|---|---|
| 31:4 | Reserved. |
| 3:0 | **StrapBifLinkConfig**. Read-write; strap. Reset: Product-specific. BIOS: See Table 35 and Table 36.<br>Bits Definition Bits Definition<br>0000b x16 IO Link (Gfx Only) 0100b 4 x1 IO Links (GPPFCH Only)<br>0001b x4 IO Link (GPPFCH Only) 0101b 2 x8 IO Links (Gfx Only)<br>0010b 2 x2 IO Links (GPPFCH Only) 011xb Reserved<br>0011b 1 x2 IO Link, 2 x1 IO Links (GPPFCH Only) 1xxxb Reserved |

### D0F0xE4_x013[1:0]_0[C:8]00 Link Training Control

**Table 66: Index address mapping for** D0F0xE4_x013[1:0]_0[C:8]00

| Index | Function | Reset | Index | Function | Reset |
|---|---|---|---|---|---|
| 0130_0800h | Device 8 | 0000_0000h | 0130_0C00h | Device 7 | 0000_0001h |
| 0130_0900h | Device 4 | 0000_0001h | 0131_0800h | Device 2 | 0000_0001h |
| 0130_0A00h | Device 5 | 0000_0001h | 0131_0800h | Device 3 | 0000_0001h |
| 0130_0B00h | Device 6 | 0000_0001h | - | - | - |

| Bits | Description |
|---|---|
| 31:1 | Reserved. |
| 0 | **HoldTraining: hold link training**. Read-write. 1=Hold training on link. |

### D0F0xE4_x013[1:0]_0[C:8]03 Link Deemphasis Control

**Table 67: Index address mapping for** D0F0xE4_x013[1:0]_0[C:8]03

| Index | Function | Index | Function |
|---|---|---|---|
| 0130_0803h | Device 8 | 0130_0C03h | Device 7 |
| 0130_0903h | Device 4 | 0131_0803h | Device 2 |
| 0130_0A03h | Device 5 | 0131_0903h | Device 3 |
| 0130_0B03h | Device 6 | - | - |

| Bits | Description |
|---|---|
| 31:6 | Reserved. |
| 5 | **StrapBifDeemphasisSel**. Read-write; strap. Reset: 1. Controls the default value of D[8:2]F0x88[SelectableDeemphasis]. |
| 4:0 | Reserved. |

### D0F0xE4_x013[2:0]_8002 IO Link Wrapper Scratch

Cold reset: 0000_0000h.
IF ((REG==D0F0xE4_x0130_8002 || REG==D0F0xE4_x0131_8002)) THEN

| Bits | Description |
|---|---|
| 31:16 | **SubsystemID: subsystem id**. Read-write. Specifies the value returned by D[8:2]F0xB4[SubsystemID]. |
| 15:0 | **SubsystemVendorID: subsystem vendor id** . Read-write. Specifies the value returned by D[8:2]F0xB4[SubsystemVendorID]. |

ELSE

| Bits | Description |
|---|---|
| 31:0 | **PcieWrapScratch: Scratch**. Read-write. |

ENDIF

**D0F0xE4_x013[2:0]_8011 Link Transmit Clock Gating Control**

| Bits | Description |
|---|---|
| 31 | **StrapBifValid**. Read-write. IF (REG==D0F0xE4_x0130_8011) THEN Reset: 0. ELSE Reset: 1. ENDIF. 0=Straps are latched. 1=Straps are not latched. |
| 30:25 | Reserved. |
| 24 | **TxclkLcntGateEnable**. Read-write. Reset: 0. BIOS: 1. 1=Enable clock gating the lane counter. See 2.5.5.3 [GPU and Root Complex Clock Gating]. |
| 23 | Reserved. |
| 22:17 | **TxclkPermGateLatency**. Read-write. Reset: 3Fh. Specifies the number of clocks to wait after detecting an entry into L1 before gating off the permanent clock branches. See 2.5.5.3 [GPU and Root Complex Clock Gating]. |
| 16 | **RcvrDetClkEnable**. Read-write. Reset: 0. 1=Enable the receiver detect clock. |
| 15:10 | **TxclkRegsGateLatency**. Read-write. Reset: 3Fh. Specifies the number of clocks to wait after idle is signaled before gating off the register clock branch. See 2.5.5.3 [GPU and Root Complex Clock Gating]. |
| 9 | **TxclkRegsGateEnable**. Read-write. Reset: 0. BIOS: 1. 1=Enable clock gating the register clock. See 2.5.5.3 [GPU and Root Complex Clock Gating]. |
| 8 | **TxclkPermStop**. Read-write. Reset: 0. 1=All transmitter clocks disabled. This bit should only be set if all links associated with the PCIe core are unconnected. See 2.5.5.3 [GPU and Root Complex Clock Gating]. |
| 7 | **TxclkDynGateEnable**. Read-write. Reset: 0. BIOS: 1. 1=Dynamic clock gating enabled. 0=Dynamic clock gating disabled. See 2.5.5.3 [GPU and Root Complex Clock Gating]. |
| 6 | **TxclkPermGateEven**. Read-write. Reset: 1. 1=Gate the permanent clock branches for an even number of clocks. See 2.5.5.3 [GPU and Root Complex Clock Gating]. |
| 5:0 | **TxclkDynGateLatency**. Read-write. Reset: 3Fh. Specifies the number of clocks to wait after idle is signaled before gating off the dynamic clock branch. See 2.5.5.3 [GPU and Root Complex Clock Gating]. |

**D0F0xE4_x013[2:0]_8012 Idle Resume Clock Gating Control**

See 2.5.5.3 [GPU and Root Complex Clock Gating].

| Bits | Description |
|---|---|
| 31:30 | Reserved. |
| 29:24 | **Pif2p5xIdleResumeLatency**. Read-write. Reset: 0_0111b. Specifies the number of clocks to wait after enabling TXCLK2P5X_PIF before sending the acknowledge. |
| 23 | **Pif2p5xIdleGateEnable**. Read-write. Reset: 0. BIOS: 1. 1=Enable idle resume gating of TXCLK2P5X_PIF. |
| 22 | Reserved. |
| 21:16 | **Pif2p5xIdleGateLatency**. Read-write. Reset: 0_0001b. Specifies the number of clocks to wait before turning off TXCLK2P5X_PIF. |
| 15:14 | Reserved. |

| | |
|---|---|
| 13:8 | **Pif1xIdleResumeLatency**. Read-write. Reset: 0_0111b. Specifies the number of clocks to wait after enabling TXCLK1X_PIF before sending the acknowledge. |
| 7 | **Pif1xIdleGateEnable**. Read-write. Reset: 0. BIOS: 1. 1=Enable idle resume gating of TXCLK1X_PIF. |
| 6 | Reserved. |
| 5:0 | **Pif1xIdleGateLatency**. Read-write. Reset: 0_0001b. Specifies the number of clocks to wait before turning off TXCLK1X_PIF. |

### D0F0xE4_x013[2:0]_8013 Transmit Clock Pll Control

Reset: 0000_0001h.

**Table 68: Reserved field mappings for** D0F0xE4_x013[2:0]_8013

| Register | Bits | | | | | | |
|---|---|---|---|---|---|---|---|
| | 20:13 | 12:11 | 10 | 7:6 | 5 | 3:2 | 1 |
| D0F0xE4_x0130_8013 | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved |
| D0F0xE4_x0131_8013 | - | - | - | - | - | - | - |
| D0F0xE4_x0132_8013 | - | Reserved | - | Reserved | - | Reserved | - |

| Bits | Description |
|---|---|
| 31:21 | Reserved. |
| 20 | **TxclkSelDigBOverride**. Read-write.1=Override TxclkDigB selection. |
| 19:17 | **TxclkSelDigB**. Read-write. Specifies the source of the dig B clock when TxclkSelDigBOverride=1.<br>Bits Definition Bits Definition<br>0x0b Phy Clock A 10xb Phy Clock C<br>0x1b Phy Clock B 11xb Phy Clock D<br>The selected clock with not function correctly if the divider logic is not enabled for the clock. The divider is enabled if (ClkDividerResetOverrideX \| MasterPciePllX \| D0F0xE4_x013[2:1]_8040[OwnPhyX]) =1. |
| 16 | **TxclkSelDigAOverride**. Read-write.1=Override TxclkDigA selection. |
| 15:13 | **TxclkSelDigA**. Read-write. Specifies the source of the dig A clock when TxclkSelDigAOverride=1.<br>Bits Definition Bits Definition<br>0x0b Phy Clock A 10xb Phy Clock C<br>0x1b Phy Clock B 11xb Phy Clock D<br>The selected clock with not function correctly if the divider logic is not enabled for the clock. The divider is enabled if (ClkDividerResetOverrideX \| MasterPciePllX \| D0F0xE4_x013[2:1]_8040[OwnPhyX]) =1. |
| 12 | **TxclkSelPifDOverride**. Read-write.1=Override TxclkPifD selection. |
| 11 | **TxclkSelPifCOverride**. Read-write. 1=Override TxclkPifC selection. |
| 10 | **TxclkSelPifBOverride**. Read-write. 1=Override TxclkPifB selection. |
| 9 | **TxclkSelPifAOverride**. Read-write. 1=Override TxclkPifA selection. |
| 8 | **TxclkSelCoreOverride**. Read-write. 1=Override TxclkCore selection. |
| 7 | **ClkDividerResetOverrideD**. Read-write. 1=Force clock divider D enabled. |
| 6 | **ClkDividerResetOverrideC**. Read-write. 1=Force clock divider C enabled. |

| | |
|---|---|
| 5 | **ClkDividerResetOverrideB**. Read-write. 1=Force clock divider B enabled. |
| 4 | **ClkDividerResetOverrideA**. Read-write. 1=Force clock divider A enabled. |
| 3 | **MasterPciePllD**. Read-write. 1=Pll D is the master source for all PCIe transmitter clock branches. |
| 2 | **MasterPciePllC**. Read-write. 1=Pll C is the master source for all PCIe transmitter clock branches. |
| 1 | **MasterPciePllB**. Read-write. 1=Pll B is the master source for all PCIe transmitter clock branches. |
| 0 | **MasterPciePllA**. Read-write. 1=Pll A is the master source for all PCIe transmitter clock branches. |

### D0F0xE4_x013[2:0]_8014 Link Transmit Clock Gating Control 2

Reset: 0000_0000h. See 2.5.5.3 [GPU and Root Complex Clock Gating].

**Table 69: Reserved field mappings for** D0F0xE4_x013[2:0]_8014

| Register | Bits | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 19:18 | 17 | 15:14 | 13 | 11:10 | 9:6 | 5:4 | 3:2 |
| D0F0xE4_x0130_8014 | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved |
| D0F0xE4_x0131_8014 | - | - | - | - | - | - | - | - |
| D0F0xE4_x0132_8014 | Reserved | - | Reserved | - | Reserved | - | Reserved | - |

| Bits | Description |
|---|---|
| 31:21 | Reserved. |
| 20 | **TxclkPermGateOnlyWhenPllPwrDn**. Read-write. BIOS: 1. 1=Gating of the permanent clock branch only occurs when the PLL is powered down. |
| 19 | **PcieGatePifD2p5xEnable**. Read-write. BIOS: 1. 1=Enable gating of the PIF D 2.5x clock branches in PCIe mode. |
| 18 | **PcieGatePifC2p5xEnable**. Read-write. BIOS: 1. 1=Enable gating of the PIF C 2.5x clock branches in PCIe mode. |
| 17 | **PcieGatePifB2p5xEnable**. Read-write. BIOS: 1. 1=Enable gating of the PIF B 2.5x clock branches in PCIe mode. |
| 16 | **PcieGatePifA2p5xEnable**. Read-write. BIOS: 1. 1=Enable gating of the PIF A 2.5x clock branches in PCIe mode. |
| 15 | **PcieGatePifD1xEnable**. Read-write. BIOS: 1. 1=Enable gating of the PIF D 1x clock branches in PCIe mode. |
| 14 | **PcieGatePifC1xEnable**. Read-write. BIOS: 1. 1=Enable gating of the PIF C 1x clock branches in PCIe mode. |
| 13 | **PcieGatePifB1xEnable**. Read-write. BIOS: 1. 1=Enable gating of the PIF B 1x clock branches in PCIe mode. |
| 12 | **PcieGatePifA1xEnable**. Read-write. BIOS: 1. 1=Enable gating of the PIF A 1x clock branches in PCIe mode. |
| 11 | **DdiGatePifD2p5xEnable**. Read-write. BIOS: 1. 1=Enable gating of the PIF D 2.5x clock branches in DDI mode. |
| 10 | **DdiGatePifC2p5xEnable**. Read-write. BIOS: 1. 1=Enable gating of the PIF C 2.5x clock branches in DDI mode. |

| | |
|---|---|
| 9 | **DdiGatePifB2p5xEnable**. Read-write. BIOS: 1. 1=Enable gating of the PIF B 2.5x clock branches in DDI mode. |
| 8 | **DdiGatePifA2p5xEnable**. Read-write. BIOS: 1. 1=Enable gating of the PIF A 2.5x clock branches in DDI mode. |
| 7 | **DdiGateDigBEnable**. Read-write. BIOS: 1. 1=Enable gating of the dig b clock branches in DDI mode. |
| 6 | **DdiGateDigAEnable**. Read-write. BIOS: 1. 1=Enable gating of the dig a clock branches in DDI mode. |
| 5 | **DdiGatePifD1xEnable**. Read-write. BIOS: 1. 1=Enable gating of the PIF D 1x clock branches in DDI mode. |
| 4 | **DdiGatePifC1xEnable**. Read-write. BIOS: 1. 1=Enable gating of the PIF C 1x clock branches in DDI mode. |
| 3 | **DdiGatePifB1xEnable**. Read-write. BIOS: 1. 1=Enable gating of the PIF B 1x clock branches in DDI mode. |
| 2 | **DdiGatePifA1xEnable**. Read-write. BIOS: 1. 1=Enable gating of the PIF A 1x clock branches in DDI mode. |
| 1 | **TxclkPrbsGateEnable**. Read-write. BIOS: 1. 1=Enable gating of the PRBS clock branch. |
| 0 | **TxclkPermGateEnable**. Read-write. BIOS: 1. 1=Enable gating of the permanent clock branch. |

### D0F0xE4_x013[2:0]_8015 IO Link IOC Control

| Bits | Description |
|---|---|
| 31 | **RefclkBphyGateEnable**. Read-write. Reset:0. BIOS: 1. 1=Enable gating of REFCLK_BPHY. See 2.5.5.3 [GPU and Root Complex Clock Gating]. |
| 30 | Reserved. |
| 29:24 | **RefclkBphyGateLatency**. Read-write. Reset:3Fh. BIOS: 0. Specifies the number of clocks to wait before turning of f REFCLK_BPHY. See 2.5.5.3 [GPU and Root Complex Clock Gating]. |
| 23 | **RefclkRegsGateEnable**. Read-write. Reset:0. BIOS: 1. 1=Enable gating of REFCLK_REGS. See 2.5.5.3 [GPU and Root Complex Clock Gating]. |
| 22 | Reserved. |
| 21:16 | **RefclkRegsGateLatency**. Read-write. Reset:3Fh. Specifies the number of clocks to wait before turning of f REFCLK_REGS. See 2.5.5.3 [GPU and Root Complex Clock Gating]. |
| 15:0 | Reserved. |

### D0F0xE4_x013[2:0]_8016 Link Clock Switching Control

Reset: 003F_001Fh.

| Bits | Description |
|---|---|
| 31:24 | Reserved. |
| 23 | **LclkDynGateEnable**. Read-write. IF (REG== D0F0xE4_x013[1:0]_8016) THEN BIOS: 1. ENDIF. 1=Enable LCLK_DYN clock gating. See 2.5.5.3 [GPU and Root Complex Clock Gating]. |

| | |
|---|---|
| 22 | **LclkGateFree**. Read-write. IF (REG== D0F0xE4_x013[1:0]_8016) THEN BIOS: 1. ENDIF. 1=LCLK gating is controlled independent of TXCLK gating. See 2.5.5.3 [GPU and Root Complex Clock Gating]. |
| 21:16 | **LclkDynGateLatency**. Read-write. Specifies the number of clocks to wait before turning off LCLK_DYN. See 2.5.5.3 [GPU and Root Complex Clock Gating]. |
| 15:6 | Reserved. |
| 5:0 | **CalibAckLatency**. Read-write. IF (REG== D0F0xE4_x0130_8016) THEN BIOS: 0. ELSEIF (REG== D0F0xE4_x0131_8016) THEN BIOS: 1. ELSE BIOS: 1Fh. ENDIF. Specifies the number of clocks after calibration is complete before the acknowledge signal is asserted. |

**D0F0xE4_x013[2:0]_8021 Transmitter Lane Mux**

Reset: 7654_3210h.

| Bits | Description |
|---|---|
| 31:28 | **Lanes1514**. Read-write. Specifies the controller lanes that are mapped to PIF TX lanes 15 and 14. See: Lanes10. |
| 27:24 | **Lanes1312**. Read-write. Specifies the controller lanes that are mapped to PIF TX lanes 13 and 12. See: Lanes10. |
| 23:20 | **Lanes1110**. Read-write. Specifies the controller lanes that are mapped to PIF TX lanes 11 and 10. See: Lanes10. |
| 19:16 | **Lanes98**. Read-write. Specifies the controller lanes that are mapped to PIF TX lanes 9 and 8. See: Lanes10. |
| 15:12 | **Lanes76**. Read-write. Specifies the controller lanes that are mapped to PIF TX lanes 7 and 6. See: Lanes10. |
| 11:8 | **Lanes54**. Read-write. Specifies the controller lanes that are mapped to PIF TX lanes 5 and 4. See: Lanes10. |
| 7:4 | **Lanes32**. Read-write. Specifies the controller lanes that are mapped to PIF TX lanes 3 and 2. See: Lanes10. |
| 3:0 | **Lanes10**. Read-write. Specifies the controller lanes that are mapped to PIF TX lanes 1 and 0.<br>Bits / Definition: 0h Controller lanes 1 and 0; 1h Controller lanes 3 and 2.; 2h Controller lanes 5 and 4.; 3h Controller lanes 7 and 6.; 4h Controller lanes 9 and 8.; 5h Controller lanes 11 and 10.; 6h Controller lanes 13 and 12.; 7h Controller lanes 15 and 14.; Fh-8h Reserved. |

**D0F0xE4_x013[2:0]_8022 Receiver Lane Mux**

Reset: 7654_3210h.

| Bits | Description |
|---|---|
| 31:28 | **Lanes1514**. Read-write. Specifies the PIF RX lanes that are mapped to controller lanes 15 and 14. See: Lanes10. |
| 27:24 | **Lanes1312**. Read-write. Specifies the PIF RX lanes that are mapped to controller lanes 13 and 12. See: Lanes10. |

| | |
|---|---|
| 23:20 | **Lanes1110**. Read-write. Specifies the PIF RX lanes that are mapped to controller lanes 11 and 10. See: Lanes10. |
| 19:16 | **Lanes98**. Read-write. Specifies the PIF RX lanes that are mapped to controller lanes 9 and 8. See: Lanes10. |
| 15:12 | **Lanes76**. Read-write. Specifies the PIF RX lanes that are mapped to controller lanes 7 and 6. See: Lanes10. |
| 11:8 | **Lanes54**. Read-write. Specifies the PIF RX lanes that are mapped to controller lanes 5 and 4. See: Lanes10. |
| 7:4 | **Lanes32**. Read-write. Specifies the PIF RX lanes that are mapped to controller lanes 3 and 2. See: Lanes10. |
| 3:0 | **Lanes10**. Read-write. Specifies the PIF RX lanes that are mapped to controller lanes 1 and 0.<br>Bits: Definition<br>0h: PIF RX lanes 1 and 0<br>1h: PIF RX lanes 3 and 2.<br>2h: PIF RX lanes 5 and 4.<br>3h: PIF RX lanes 7 and 6.<br>4h: PIF RX lanes 9 and 8.<br>5h: PIF RX lanes 11 and 10.<br>6h: PIF RX lanes 13 and 12.<br>7h: PIF RX lanes 15 and 14.<br>Fh-8h: Reserved. |

### D0F0xE4_x013[2:0]_8023 Lane Enable

Reset: 0000_FFFFh.

| Bits | Description |
|---|---|
| 31:16 | Reserved. |
| 15:0 | **LaneEnable**. Read-write. 1=Lane enabled for transmit.<br>Bit: Definition<br>[15:0]: Lane <BIT> enable |

### D0F0xE4_x013[2:0]_8025 Lane Mux Power Sequence Control

Reset: 1F1F_1F1Fh.

| Bits | Description |
|---|---|
| 31:30 | Reserved. |
| 29 | **LMLinkSpeed3**. Read-write. 1=5GHz. 0=2.5GHz. |
| 28:27 | **LMRxPhyCmd3**. Read-write. Specifies the receiver state for lanes 15:12. See: LMRxPhyCmd0. |
| 26:24 | **LMTxPhyCmd3**. Read-write. Specifies the transmitter state for lanes 15:12. See: LMTxPhyCmd0. |
| 23:22 | Reserved. |
| 21 | **LMLinkSpeed2**. Read-write. 1=5GHz. 0=2.5GHz. |
| 20:19 | **LMRxPhyCmd2**. Read-write. Specifies the receiver state for lanes 11:8. See: LMRxPhyCmd0. |
| 18:16 | **LMTxPhyCmd2**. Read-write. Specifies the transmitter state for lanes 11:8. See: LMTxPhyCmd0. |
| 15:14 | Reserved. |
| 13 | **LMLinkSpeed1**. Read-write. 1=5GHz. 0=2.5GHz. |
| 12:11 | **LMRxPhyCmd1**. Read-write. Specifies the receiver state for lanes 7:4. See: LMRxPhyCmd0. |
| 10:8 | **LMTxPhyCmd1**. Read-write. Specifies the transmitter state for lanes 7:4. See: LMTxPhyCmd0. |

| | |
|---|---|
| 7:6 | Reserved. |
| 5 | **LMLinkSpeed0**. Read-write. 1=5GHz. 0=2.5GHz. |
| 4:3 | **LMRxPhyCmd0**. Read-write. Specifies the receiver state for lanes 3:0.<br>Bits Definition Bits Definition<br>00b On. 10b Standby 2 (L1).<br>01b Standby 1 (L0s). 11b Off. |
| 2:0 | **LMTxPhyCmd0**. Read-write. Specifies the transmitter state for lanes 3:0.<br>Bits Definition Bits Definition<br>000b On. 100b Receiver detect.<br>001b Standby 1 (L0s). 101b Reserved.<br>010b Standby 2 (L1). 110b Reserved.<br>011b Reserved. 111b Off. |

### D0F0xE4_x013[1:0]_8031 Lane Counter Status

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:17 | Reserved. |
| 16 | **LnCntValid**. Read-write. 1=LnCntBandwidth contains a valid bandwidth measurement. |
| 15:10 | Reserved. |
| 9:0 | **LnCntBandwidth**. Read-write. Estimated lane bandwidth in 10 MB/s units. |

### D0F0xE4_x013[2:0]_8060 Soft Reset Command 0

Cold reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:18 | Reserved. |
| 17 | **BifCalibrationReset**. Read-write. 1=The BIF calibration block reset is asserted. |
| 16 | **BifGlobalReset**. Read-write. 1=The BIF global reset is asserted. |
| 15:3 | Reserved. |
| 2 | **ResetComplete**. Read-only. 1=Reset cycle is complete. |
| 1 | Reserved. |
| 0 | **Reconfigure**. Read-write; Cleared-when-done. 1=Trigger atomic reconfiguration if D0F0xE4_x013[2:0]_8062[ReconfigureEn]=1. |

### D0F0xE4_x013[2:0]_8062 Soft Reset Control 0

Cold reset: 0001_0800h.

| Bits | Description |
|---|---|
| 31:12 | Reserved. |
| 11 | **ConfigXferMode**. Read-write. 1=PCIe core strap settings take effect immediately. 0=PCIe core strap settings take effect when the PCIe core is reset. |

| | |
|---|---|
| 10 | **BlockOnIdle**. Read-write. 1=The PCIe core must be idle before hardware initiates a reconfiguration. 0=The PCIe core does not have to be idle before hardware initiates a reconfiguration. |
| 9:5 | Reserved. |
| 4:2 | **ResetPeriod**. Read-write. BIOS:0. Specifies the amount of time that resets are asserted during a reconfiguration. |
| 1 | Reserved. |
| 0 | **ReconfigureEn**. Read-write. 1=Atomic reconfiguration enabled. |

**D0F0xE4_x0130_80F0 BIOS Timer**

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:0 | **MicroSeconds**. Read-write; updated-by-hardware. This field increments once every microsecond when the timer is enabled. The counter will roll over and continue counting when it reaches its FFFF_FFFFh. A write to this register causes the counter to reset and begin counting from the value written. |

**D0F0xE4_x0130_80F1 BIOS Timer Control**

Reset: 0000_0064h.

| Bits | Description |
|---|---|
| 31:8 | Reserved. |
| 7:0 | **ClockRate**. Read-write. Specifies the frequency of the reference clock in 1 MHz increments.<br>Bits Definition<br>00h Timer disabled<br>FFh-01h <ClockRate> MHz |

**D0F0xE4_x0130_FFF1 GPP Capabilities**

| Bits | Description |
|---|---|
| 31:8 | Reserved. |
| 7 | **ROSupportGen2**. Read-write. Cold reset: Product-specific. |
| 6 | **LcSupportGen2**. Read-write. Cold reset: Product-specific. |
| 5:0 | Reserved. |

**D0F0xE4_x013[2:1]_8040 DDI Control**

Reset: 0000_7700h.

**Table 70: Reserved field mappings for** D0F0xE4_x013[2:1]_8040

| Register | Bits | |
|---|---|---|
| | 19:18 | 3:2 |
| D0F0xE4_x0131_8040 | - | - |
| D0F0xE4_x0132_8040 | Reserved. Read-write | Reserved. Read-write |

| Bits | Description |
|---|---|
| 31:23 | Reserved. |
| 22 | **ChangeLnSpd**. Read-write. 1=Link speed is changed for all low power state lanes. |
| 21 | **CntDigB**. Read-write. 1=Enable software control over Dig B TxPhyCmd and Link Speed. |
| 20 | **CntDigA**. Read-write. 1=Enable software control over Dig A TxPhyCmd and Link Speed. |
| 19 | **CntPhyD**. Read-write. 1=Enable software control over phy D state machine. |
| 18 | **CntPhyC**. Read-write. 1=Enable software control over phy C state machine. |
| 17 | **CntPhyB**. Read-write. 1=Enable software control over phy B state machine. |
| 16 | **CntPhyA**. Read-write. 1=Enable software control over phy A state machine. |
| 15 | Reserved. Read-write. |
| 14:12 | **DigbPwrdnValue**. Read-write. Specifies the phy power state for links associated with dig stream b that are not enabled. See: DigaPwrdnValue. |
| 11 | Reserved. Read-write. |
| 10:8 | **DigaPwrdnValue**. Read-write. Specifies the phy power state for links associated with dig stream a that are not enabled.<br>Bits: 000b Definition: On.; 001b Standby 1 (L0s).; 010b Standby 2 (L1).; 011b Reserved.; 100b Receiver detect.; 101b Reserved.; 110b Reserved.; 111b Off. |
| 7:4 | Reserved. |
| 3 | **OwnPhyD**. Read-write. 1=DDI block controls lanes 15:12. |
| 2 | **OwnPhyC**. Read-write. 1=DDI block controls lanes 11:8. |
| 1 | **OwnPhyB**. Read-write. 1=DDI block controls lanes 7:4. |
| 0 | **OwnPhyA**. Read-write. 1=DDI block controls lanes 3:0. |

## 3.4 Device 1 Function 0 (Internal Graphics) Configuration Registers

See 3.1 [Register Descriptions and Mnemonics] for a description of the register naming convention. See 2.7 [Configuration Space] for details about how to access this space.

**D1F0x00 Device/Vendor ID Register**

| Bits | Description |
|---|---|
| 31:16 | **DeviceID: device ID**. Read-only. Reset: Product-specific. |
| 15:0 | **VendorID: vendor ID**. Read-only. Reset: 1002h. |

**D1F0x04 Status/Command Register**

Reset: 0010_0000h.

| Bits | Description |
|---|---|
| 31 | **ParityErrorDetected: detected parity error**. Read; Write-1-to-clear. 1=Poisoned TLP received. |
| 30 | **SignaledSystemError: signaled system error**. Read; Write-1-to-clear. 1=A non-fatal or fatal error message was sent and SerrEn=1. |
| 29 | **ReceivedMasterAbort: received master abort**. Read; Write-1-to-clear. 1=A completion with an unsupported request completion status was received. |
| 28 | **ReceivedTargetAbort: received target abort**. Read; Write-1-to-clear. 1=A completion with completer abort completion status was received. |
| 27 | **SignalTargetAbort: Signaled target abort**. Read-only. |
| 26:25 | **DevselTiming: DEVSEL# Timing**. Read-only. |
| 24 | **MasterDataPerr: master data parity error**. Read; Write-1-to-clear. 1=ParityErrorEn=1 and either a poisoned completion was received or the device poisoned a write request. |
| 23 | **FastBackCapable: fast back-to-back capable**. Read-only. |
| 22 | Reserved. |
| 21 | **PCI66En: 66 MHz capable**. Read-only. |
| 20 | **CapList: capability list**. Read-only. 1=capability list supported. |
| 19 | **IntStatus: interrupt status**. Read-only. 1=INTx interrupt message pending. |
| 18:11 | Reserved. |
| 10 | **IntDis: interrupt disable**. Read-write. 1=INTx interrupt messages generation disabled. |
| 9 | **FastB2BEn: fast back-to-back enable**. Read-only. |
| 8 | **SerrEn: System error enable**. Read-write. 1=Enables reporting of non-fatal and fatal errors detected. |
| 7 | **Stepping: Stepping control**. Read-only. |
| 6 | **ParityErrorEn: parity error response enable**. Read-write. |
| 5 | **PalSnoopEn: VGA palette snoop enable**. Read-only. |
| 4 | **MemWriteInvalidateEn: memory write and invalidate enable**. Read-only. |
| 3 | **SpecialCycleEn: special cycle enable**. Read-only. |
| 2 | **BusMasterEn: bus master enable**. Read-write. 1=Memory and IO read and write request generation enabled. |

| | |
|---|---|
| 1 | **MemAccessEn: memory access enable**. Read-write. This bit controls if memory accesses targeting this device are accepted. 1=Enabled. 0=Disabled. |
| 0 | **IoAccessEn: IO access enable**. Read-write. This bit controls if IO accesses targeting this device are accepted. 1=Enabled. 0=Disabled. |

### D1F0x08 Class Code/Revision ID Register

| Bits | Description |
|---|---|
| 31:8 | **ClassCode**. Value: 03_0000h. |
| 7:0 | **RevID: revision ID**. Value: 00h. |

### D1F0x0C Header Type Register

Reset: 0080_0000h.

| Bits | Description |
|---|---|
| 31:24 | **BIST**. Read-only. |
| 23:16 | **HeaderTypeReg**. Read-only. The header type field indicates a header type 0 and that this is a multi-function device. |
| 15:8 | **LatencyTimer**. Read-only. These bits are fixed at their default value. |
| 7:0 | **CacheLineSize**. Read-write. This field specifies the system cache line size in units of double words. |

### D1F0x10 Graphic Memory Base Address

IF (D0F0x64_x1C[F064BarEn]==0) THEN Reset: 0000_0008h. ELSE Reset: 0000_000Ch. ENDIF.

| Bits | Description |
|---|---|
| 31:26 | **BaseAddr[31:26]: base address**. Read-write. The amount of memory requested by the graphics memory BAR is controlled by D0F0x64_x1C[MemApSize]. |
| 25:4 | **BaseAddr[25:4]: base address**. Read-only. |
| 3 | **Pref: prefetchable**. Read-only. 1=Prefetchable memory region. |
| 2:1 | **Type: base address register type**. Read-only. 00b=32-bit BAR. 10b=64-bit BAR. |
| 0 | **MemSpace: memory space type**. Read-only. 0=Memory mapped base address. |

IF (D0F0x64_x1C[F064BarEn]==0) THEN

### D1F0x14 Graphics IO Base Address

Reset: 0000_0001h.

| Bits | Description |
|---|---|
| 31:8 | **BaseAddr: base address**. IF (D0F0x64_x1C[IoBarDis]==0) THEN Read-write. ELSE Read-only. ENDIF. |
| 7:1 | Reserved. |
| 0 | **MemSpace: memory space type**. Read-only. 1=IO mapped base address. |

ELSE

**D1F0x14 Graphics Memory Base Address 64**

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:0 | **BaseAddr[63:32]: base address**. Read-write. |

ENDIF.

**D1F0x18 Graphics Memory Mapped Registers Base Address**

IF (D0F0x64_x1C[F064BarEn]==0) THEN Reset: 0000_0000h. ELSE Reset: 0000_0004h. ENDIF.

| Bits | Description |
|---|---|
| 31:18 | **BaseAddr[31:18]: base address**. Read-write. |
| 17:16 | **BaseAddr[17:16]: base address**. IF (D0F0x64_x1C[RegApSize]==0) THEN Read-write. ELSE Read-only. ENDIF. |
| 15:4 | **BaseAddr[15:4]: base address**. Read-only. |
| 3 | **Pref: prefetchable**. Read-only. 0=Non-prefetchable memory region. |
| 2:1 | **Type: base address register type**. Read-only. 00b=32-bit BAR. 10b=64-bit BAR. |
| 0 | **MemSpace: memory space type**. Read-only. 0=Memory mapped base address. |

IF (D0F0x64_x1C[F064BarEn]==0) THEN

**D1F0x1C Base Address 3**

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:0 | Reserved. |

ELSE

**D1F0x1C Graphics Memory Mapped Registers Address 64**

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:0 | **BaseAddr[63:32]: base address**. Read-write. |

ENDIF.

IF (D0F0x64_x1C[F064BarEn]==0) THEN

**D1F0x20 Base Address 4**

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:0 | Reserved. |

ELSE

**D1F0x20 Graphics IO Base Address**

Reset: 0000_0001h.

| Bits | Description |
|---|---|
| 31:8 | **BaseAddr: base address**. IF (D0F0x64_x1C[IoBarDis]==0) THEN Read-write. ELSE Read-only. ENDIF. |
| 7:1 | Reserved. |
| 0 | **MemSpace: memory space type**. Read-only. 1=IO mapped base address. |

ENDIF.

**D1F0x2C Subsystem and Subvendor ID Register**

| Bits | Description |
|---|---|
| 31:16 | **SubsystemID**. Value: D1F0x4C[SubsystemID]. |
| 15:0 | **SubsystemVendorID**. Value: D1F0x4C[SubsystemVendorID]. |

**D1F0x30 Expansion ROM Base Address**

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:0 | Reserved. |

**D1F0x34 Capabilities Pointer**

Reset: 0000_0050h.

| Bits | Description |
|---|---|
| 31:8 | Reserved. |
| 7:0 | **CapPtr: capabilities pointer**. Read-only. Pointer to PM capability. |

**D1F0x3C Interrupt Line**

Reset: 0000_01FFh.

| Bits | Description |
|---|---|
| 31:16 | Reserved. |
| 15:8 | **InterruptPin: interrupt pin**. Read-only. This field identifies the legacy interrupt message the function uses. |
| 7:0 | **InterruptLine: interrupt line**. Read-write. This field contains the interrupt line routing information. |

### D1F0x4C Subsystem and Subvendor ID Mirror

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:16 | **SubsystemID**. Read-write. This field sets the value in the corresponding field in D1F0x2C. |
| 15:0 | **SubsystemVendorID**. Read-write. This field sets the value in the corresponding field in D1F0x2C. |

### D1F0x50 Power Management Capability

| Bits | Description |
|---|---|
| 31:27 | **PmeSupport**. Value: 0_0000b. Indicates that there is no PME support. |
| 26 | **D2Support: D2 support**. Value: 1. D2 is supported |
| 25 | **D1Support: D1 support**. Value: 1. D1 is supported |
| 24:22 | **AuxCurrent: auxiliary current**. Value: 0. |
| 21 | **DevSpecificInit: device specific initialization**. Value: 0. Indicates that there is no device specific initialization necessary. |
| 20 | Reserved. |
| 19 | **PmeClock**. Value: 0. |
| 18:16 | **Version: version**. Value: 011b. |
| 15:8 | **NextPtr: next pointer**. Value: 58h. |
| 7:0 | **CapID: capability ID**. Value: 01h. Indicates that the capability structure is a PCI power management data structure. |

### D1F0x54 Power Management Control and Status

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:24 | **PmeData**. Read-only. |
| 23 | **BusPwrEn**. Read-only. |
| 22 | **B2B3Support**. Read-only. B states are not supported. |
| 21:16 | Reserved. |
| 15 | **PmeStatus: PME status**. Read-only. |
| 14:13 | **DataScale: data scale**. Read-only. |
| 12:9 | **DataSelect: data select**. Read-only. |
| 8 | **PmeEn: PME# enable**. Read-only. |
| 7:4 | Reserved. |
| 3 | **NoSoftReset: no soft reset**. Read-only. Software is required to re-initialize the function when returning from $D3_{hot}$. |

| | |
|---|---|
| 2 | Reserved. |
| 1:0 | **PowerState: power state**. Read-write. This 2-bit field is used both to determine the current power state of the root port and to set the root port into a new power state.<br>Bits Definition<br>00b D0<br>11b D3hot |

**D1F0x58 PCI Express Capability**

| Bits | Description |
|---|---|
| 31:30 | Reserved. |
| 29:25 | **IntMessageNum: interrupt message number**. Value: 0. This field indicates which MSI vector is used for the interrupt message. |
| 24 | **SlotImplemented: Slot implemented**. Value: 0. |
| 23:20 | **DeviceType: device type**. Value: IF (D0F0x64_x1C[RcieEn]==1) THEN 9. ELSEIF (D0F0x64_x1C[F0NonlegacyDeviceTypeEn]==0) THEN 1. ELSE 0. ENDIF. |
| 19:16 | **Version**. Value: 2h. |
| 15:8 | **NextPtr: next pointer**. Value: IF (D0F0x64_x1C[MsiDis]==0) THEN A0h. ELSE 00h. ENDIF. |
| 7:0 | **CapID: capability ID**. Value: 10h. |

**D1F0x5C Device Capability**

| Bits | Description |
|---|---|
| 31:29 | Reserved. |
| 28 | **FlrCapable: function level reset capability**. Value: 0. |
| 27:26 | **CapturedSlotPowerScale: captured slot power limit scale**. Value: 0. |
| 25:18 | **CapturedSlotPowerLimit: captured slot power limit value**. Value: 0. |
| 17:16 | Reserved. |
| 15 | **RoleBasedErrReporting: role-based error reporting**. Value: 1. |
| 14:12 | Reserved. |
| 11:9 | **L1AcceptableLatency: endpoint L1 Acceptable Latency**. Value: 111b. |
| 8:6 | **L0SAcceptableLatency: endpoint L0s Acceptable Latency**. Value: 110b. |
| 5 | **ExtendedTag: extended tag support**. Value: 1. 8 bit tag support. |
| 4:3 | **PhantomFunc: phantom function support**. Value: 0. No phantom functions supported. |
| 2:0 | **MaxPayloadSupport: maximum supported payload size**. Value: 000b. 128 bytes max payload size. |

**D1F0x60 Device Control and Status**

Reset: 0000_0810h.

| Bits | Description |
|---|---|
| 31:22 | Reserved. |
| 21 | **TransactionsPending: transactions pending**. Read-only. |
| 20 | **AuxPwr: auxiliary power**. Read-only. |
| 19 | **UsrDetected: unsupported request detected**. Read; Write-1-to-clear. 1=Unsupported request received. |
| 18 | **FatalErr: fatal error detected**. Read; Write-1-to-clear. 1=Fatal error detected. |
| 17 | **NonFatalErr: non-fatal error detected**. Read; Write-1-to-clear. 1=Non-fatal error detected. |
| 16 | **CorrErr: correctable error detected**. Read; Write-1-to-clear. 1=Correctable error detected. |
| 15 | **BridgeCfgRetryEn: bridge configuration retry enable**. Read-only. |
| 14:12 | **MaxRequestSize: maximum request size**. Read-only. |
| 11 | **NoSnoopEnable: enable no snoop**. Read-write. 1=The device is permitted to set the No Snoop bit in requests. |
| 10 | **AuxPowerPmEn: auxiliary power PM enable**. Read-only. This capability is not implemented. |
| 9 | **PhantomFuncEn: phantom functions enable**. Read-only. Phantom functions are not supported. |
| 8 | **ExtendedTagEn: extended tag enable**. Read-write. 1=8-bit tag request tags. 0=5-bit request tag. |
| 7:5 | **MaxPayloadSize: maximum supported payload size**. Read-only. 000b=Indicates a 128 byte maximum payload size. |
| 4 | **RelaxedOrdEn: relaxed ordering enable**. Read-write. 1=The device is permitted to set the Relaxed Ordering bit. |
| 3 | **UsrReportEn: unsupported request reporting enable**. Read-write. 1=Enables signaling unsupported requests by sending error messages. |
| 2 | **FatalErrEn: fatal error reporting enable**. Read-write. 1=Enables sending ERR_FATAL message when a fatal error is detected. |
| 1 | **NonFatalErrEn: non-fatal error reporting enable**. Read-write. 1=Enables sending ERR_NONFATAL message when a non-fatal error is detected. |
| 0 | **CorrErrEn: correctable error reporting enable**. Read-write. 1=Enables sending ERR_CORR message when a correctable error is detected. |

**D1F0x64 Link Capability**

| Bits | Description |
|---|---|
| 31:24 | **PortNumber: port number**. Value: 0.This field indicates the PCI Express port number for the given PCI Express link. |
| 23:22 | Reserved. |
| 21 | **LinkBWNotificationCap: link bandwidth notification capability**. Value: 0b. |
| 20 | **DlActiveReportingCapable: data link layer active reporting capability**. Value: 0b. |

| | |
|---|---|
| 19 | **SurpriseDownErrReporting: surprise down error reporting capability**. Value: 0b. |
| 18 | **ClockPowerManagement: clock power management**. Value: 0b. |
| 17:15 | **L1ExitLatency: L1 exit latency**. Value: 000b. |
| 14:12 | **L0sExitLatency: L0s exit latency**. Value: 000b. |
| 11:10 | **PMSupport: active state power management support**. Value: 00b. |
| 9:4 | **LinkWidth: maximum link width**. Value: 0. |
| 3:0 | **LinkSpeed: link speed**. Value: 0. |

**D1F0x68 Link Control and Status**

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31 | **LinkAutonomousBWStatus: link autonomous bandwidth status**. Read-only. |
| 30 | **LinkBWManagementStatus: link bandwidth management status**. Read-only. |
| 29 | **DlActive: data link layer link active**. Read-only. This bit indicates the status of the data link control and management state machine. Reads return a 1 to indicate the DL_Active state, otherwise 0 is returned. |
| 28 | **SlotClockCfg: slot clock configuration**. Read-only. 1=The root port uses the same clock that the platform provides. |
| 27 | **LinkTraining: link training**. Read-only. 1=Indicates that the physical layer link training state machine is in the configuration or recovery state, or that 1b was written to the RetrainLink bit but link training has not yet begun. Hardware clears this bit when the link training state machine exits the configuration/recovery state. |
| 26 | Reserved. |
| 25:20 | **NegotiatedLinkWidth: negotiated link width**. Read-only. This field indicates the negotiated width of the given PCI Express link. |
| 19:16 | **LinkSpeed: link speed**. Read-only. |
| 15:12 | Reserved. |
| 11 | **LinkAutonomousBWIntEn: link autonomous bandwidth interrupt enable**. Read-only. |
| 10 | **LinkBWManagementEn: link bandwidth management interrupt enable**. Read-only. |
| 9 | **HWAutonomousWidthDisable: hardware autonomous width disable**. Read-write. 1=Hardware not allowed to change the link width except to correct unreliable link operation by reducing link width. |
| 8 | **ClockPowerManagementEn: clock power management enable**. Read-write. |
| 7 | **ExtendedSync: extended sync**. Read-write. 1=Forces the transmission of additional ordered sets when exiting the L0s state and when in the recovery state. |
| 6 | **CommonClockCfg: common clock configuration**. Read-write. 1=Indicates that the root port and the component at the opposite end of this Link are operating with a distributed common reference clock. 0=Indicates that the upstream port and the component at the opposite end of this Link are operating with asynchronous reference clock. |
| 5 | **RetrainLink: retrain link**. Read-only. This bit does not apply to endpoints. |
| 4 | **LinkDis: link disable**. Read-only. This bit does not apply to endpoints. |

| | |
|---|---|
| 3 | **ReadCplBoundary: read completion boundary**. Read-only. 0=64 byte read completion boundary. |
| 2 | Reserved. |
| 1:0 | **PmControl: active state power management enable**. Read-write. This field controls the level of ASPM supported on the given PCI Express link.<br>Bits Definition Bits Definition<br>00b Disabled. 10b L1 Entry Enabled.<br>01b L0s Entry Enabled. 11b L0s and L1 Entry Enabled. |

**D1F0x7C Device Capability 2**

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:5 | Reserved. |
| 4 | **CplTimeoutDisSup: completion timeout disable supported**. Read-only. |
| 3:0 | **CplTimeoutRangeSup: completion timeout range supported**. Read-only. |

**D1F0x80 Device Control and Status 2**

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:5 | Reserved. |
| 4 | **CplTimeoutDis: completion timeout disable**. Read-only. |
| 3:0 | **CplTimeoutValue: completion timeout range supported**. Read-only. |

**D1F0x84 Link Capability 2**

| Bits | Description |
|---|---|
| 31:0 | Reserved. |

**D1F0x88 Link Control and Status 2**

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:17 | Reserved. |
| 16 | **CurDeemphasisLevel: current deemphasis level**. Read-only. 1=-3.5 dB. 0=-6 dB. |
| 15:13 | Reserved. |
| 12 | **ComplianceDeemphasis: compliance deemphasis**. Read-write. This bit defines the deemphasis level used in compliance mode. 1=-3.5 dB. 0=-6 dB. |
| 11 | **ComplianceSOS: compliance SOS**. Read-write. 1=The device transmits skip ordered sets in between the modified compliance pattern. |

| | |
|---|---|
| 10 | **EnterModCompliance: enter modified compliance**. Read-write. 1=The device transmits modified compliance pattern. |
| 9:7 | **XmitMargin: transmit margin**. Read-write. This field controls the non-deemphasized voltage level at the transmitter pins. |
| 6 | **SelectableDeemphasis: selectable deemphasis**. Read-only. |
| 5 | **HwAutonomousSpeedDisable: hardware autonomous speed disable**. Read-write. 1=Disables hardware generated link speed changes. |
| 4 | **EnterCompliance: enter compliance**. Read-write. 1=Force link to enter compliance mode. |
| 3:0 | **TargetLinkSpeed: target link speed**. Read-write. This fields defines the upper limit of the link operational speed. |

### D1F0xA0 MSI Capability

| Bits | Description |
|---|---|
| 31:24 | Reserved. |
| 23 | **Msi64bit: MSI 64 bit capability**. Read-only. Value: D0F0x64_x46[Msi64bitEn]. 1=The device is capable of sending 64-bit MSI messages. 0=The device is not capable of sending a 64-bit message address. |
| 22:20 | **MsiMultiEn: MSI multiple message enable**. Read-write. Reset: 000b. Software writes to this field to indicate the number of allocated vectors (equal to or less than the number of requested vectors). When MSI is enabled, a function is allocated at least 1 vector. |
| 19:17 | **MsiMultiCap: MSI multiple message capability**. Read-only. Reset:000b. 000b=The device is requesting one vector. |
| 16 | **MsiEn: MSI enable**. Read-write. Reset: 0. 1=MSI generation is enabled and INTx generation is disabled. 0=MSI generation disabled and INTx generation is enabled. |
| 15:8 | **NextPtr: next pointer**. Read-only. Reset:00h. |
| 7:0 | **CapID: capability ID**. Read-only. Reset: 05h. 05h=MSI capability structure. |

### D1F0xA4 MSI Message Address Low

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:2 | **MsiMsgAddrLo: MSI message address**. Read-write. This register specifies the dword aligned address for the MSI memory write transaction. |
| 1:0 | Reserved. |

IF (D0F0x64_x46[Msi64bitEn]==1) THEN

**D1F0xA8 MSI Message Address High**

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:8 | Reserved. |
| 7:0 | **MsiMsgAddrHi: MSI message address**. Read-write. This register specifies the upper 8-bits of the MSI address in 64 bit MSI mode. |

ENDIF.
IF (D0F0x64_x46[Msi64bitEn]==0) THEN

**D1F0xA8 MSI Message Data**

ELSEIF (D0F0x64_x46[Msi64bitEn]==1) THEN

**D1F0xAC MSI Message Data**

ENDIF.
Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:16 | Reserved. |
| 15:0 | **MsiData: MSI message data**. Read-write. This register specifies lower 16 bits of data for the MSI memory write transaction. The upper 16 bits are always 0. |

**D1F0x100 Vendor Specific Enhanced Capability**

Reset: 0001_000Bh.

| Bits | Description |
|---|---|
| 31:20 | **NextPtr: next pointer**. Read-only. |
| 19:16 | **CapVer: capability version**. Read-only. |
| 15:0 | **CapID: capability ID**. Read-only. |

**D1F0x104 Vendor Specific Header**

Reset: 0101_0001h.

| Bits | Description |
|---|---|
| 31:20 | **VsecLen: vendor specific enhanced next pointer**. Read-only. |
| 19:16 | **VsecRev: vendor specific enhanced capability version**. Read-only. |
| 15:0 | **VsecID: vendor specific enhanced capability ID**. Read-only. |

**D1F0x108 Vendor Specific 1**

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:0 | **Scratch: scratch**. Read-write. |

**D1F0x10C Vendor Specific 2**

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:0 | **Scratch: scratch**. Read-write. |

## 3.5 Device 1 Function 1 (Audio Controller) Configuration Registers

Access to the internal graphic configuration registers requires that internal graphic bridge is configured to forward configuration transactions to the graphics bus. See 3.1 [Register Descriptions and Mnemonics] for a description of the register naming convention. See 2.7 [Configuration Space] for details about how to access this space.

**D1F1x00 Device/Vendor ID**

| Bits | Description |
|---|---|
| 31:16 | **DeviceID: device ID**. Read-only. Reset: 1714h. |
| 15:0 | **VendorID: vendor ID**. Read-only. Reset: 1002h. |

**D1F1x04 Status/Command**

Reset: 0010_0000h.

| Bits | Description |
|---|---|
| 31 | **ParityErrorDetected: detected parity error**. Read; Write-1-to-clear. 1=Poisoned TLP received. |
| 30 | **SignaledSystemError: signaled system error**. Read; Write-1-to-clear. 1=A non-fatal or fatal error message was sent and SerrEn=1. |
| 29 | **ReceivedMasterAbort: received master abort**. Read; Write-1-to-clear. 1=A completion with an unsupported request completion status was received. |
| 28 | **ReceivedTargetAbort: received target abort**. Read; Write-1-to-clear. 1=A completion with completer abort completion status was received. |
| 27 | **SignalTargetAbort: Signaled target abort**. Read-only. |
| 26:25 | **DevselTiming: DEVSEL# Timing**. Read-only. |
| 24 | **MasterDataPerr: master data parity error**. Read; Write-1-to-clear. 1=ParityErrorEn=1 and either a poisoned completion was received or the device poisoned a write request. |
| 23 | **FastBackCapable: fast back-to-back capable**. Read-only. |
| 22 | Reserved. |

| | |
|---|---|
| 21 | **PCI66En: 66 MHz capable**. Read-only. |
| 20 | **CapList: capability list**. Read-only. 1=capability list supported. |
| 19 | **IntStatus: interrupt status**. Read-only. 1=INTx interrupt message pending. |
| 18:11 | Reserved. |
| 10 | **IntDis: interrupt disable**. Read-write. 1=INTx interrupt messages generation disabled. |
| 9 | **FastB2BEn: fast back-to-back enable**. Read-only. |
| 8 | **SerrEn: System error enable**. Read-write. 1=Enables reporting of non-fatal and fatal errors detected. |
| 7 | **Stepping: Stepping control**. Read-only. |
| 6 | **ParityErrorEn: parity error response enable**. Read-write. |
| 5 | **PalSnoopEn: VGA palette snoop enable**. Read-only. |
| 4 | **MemWriteInvalidateEn: memory write and invalidate enable**. Read-only. |
| 3 | **SpecialCycleEn: special cycle enable**. Read-only. |
| 2 | **BusMasterEn: bus master enable**. Read-write. 1=Memory and IO read and write request generation enabled. |
| 1 | **MemAccessEn: memory access enable**. Read-write. This bit controls if memory accesses targeting this device are accepted. 1=Enabled. 0=Disabled. |
| 0 | **IoAccessEn: IO access enable**. Read-write. This bit controls if IO accesses targeting this device are accepted. 1=Enabled. 0=Disabled. |

### D1F1x08 Class Code/Revision ID

Reset: 0403_0000h.

| Bits | Description |
|---|---|
| 31:8 | **ClassCode**. Read-only. |
| 7:0 | **RevID: revision ID**. Read-only. |

### D1F1x0C Header Type

Reset: 0080_0000h.

| Bits | Description |
|---|---|
| 31:24 | **BIST**. Read-only. These bits are fixed at their default values. |
| 23:16 | **HeaderTypeReg**. Read-only. 80h=Type 0 multi-function device. |
| 15:8 | **LatencyTimer**. Read-only. These bits are fixed at their default value. |
| 7:0 | **CacheLineSize**. Read-write. This field specifies the system cache line size in units of double words. |

**D1F1x10 Audio Registers Base Address**

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:14 | **BaseAddr: base address**. Read-write. |
| 13:4 | Reserved. |
| 3 | **Pref: prefetchable**. Read-only. 0=Non-prefetchable memory region. |
| 2:1 | **Type: base address register type**. Read-only. 00b=32-bit base address register. |
| 0 | **MemSpace: memory space type**. Read-only. 0=Memory mapped base address. |

**D1F1x14 Base Address 1**

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:0 | Reserved. |

**D1F1x18 Base Address 2**

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:0 | Reserved. |

**D1F1x1C Base Address 3**

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:0 | Reserved. |

**D1F1x20 Base Address 4**

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:0 | Reserved. |

**D1F1x24 Base Address 5**

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:0 | Reserved. |

### D1F1x2C Subsystem and Subvendor ID

| Bits | Description |
|---|---|
| 31:16 | **SubsystemID**. Value: D1F1x4C[SubsystemID]. |
| 15:0 | **SubsystemVendorID**. Value: D1F1x4C[SubsystemVendorID]. |

### D1F1x30 Expansion ROM Base Address

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:0 | Reserved. |

### D1F1x34 Capabilities Pointer

Reset: 0000_0050h.

| Bits | Description |
|---|---|
| 31:8 | Reserved. |
| 7:0 | **CapPtr: capabilities pointer**. Read-only. Pointer to PM capability. |

### D1F1x3C Interrupt Line

Reset: 0000_02FFh.

| Bits | Description |
|---|---|
| 31:16 | Reserved. |
| 15:8 | **InterruptPin: interrupt pin**. Read-only. This field identifies the legacy interrupt message the function uses. |
| 7:0 | **InterruptLine: interrupt line**. Read-write. This field contains the interrupt line routing information. |

### D1F1x4C Subsystem and Subvendor ID Mirror

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:16 | **SubsystemID**. Read-write. This field sets the value in the corresponding field in D1F1x2C. |
| 15:0 | **SubsystemVendorID**. Read-write. This field sets the value in the corresponding field in D1F1x2C. |

### D1F1x50 Power Management Capability

| Bits | Description |
|---|---|
| 31:27 | **PmeSupport**. Value: 0_0000b. Indicates that there is no PME support. |

| | |
|---|---|
| 26 | **D2Support: D2 support**. Value: 1. D2 is supported |
| 25 | **D1Support: D1 support**. Value: 1. D1 is supported |
| 24:22 | **AuxCurrent: auxiliary current**. Value: 0. |
| 21 | **DevSpecificInit: device specific initialization**. Value: 0. Indicates that there is no device specific initialization necessary. |
| 20 | Reserved. |
| 19 | **PmeClock**. Value: 0. |
| 18:16 | **Version: version**. Value: 011b. |
| 15:8 | **NextPtr: next pointer**. Value: 58h. |
| 7:0 | **CapID: capability ID**. Value: 01h. Indicates that the capability structure is a PCI power management data structure. |

### D1F1x54 Power Management Control and Status

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:24 | **PmeData**. Read-only. |
| 23 | **BusPwrEn**. Read-only. |
| 22 | **B2B3Support**. Read-only. B states are not supported. |
| 21:16 | Reserved. |
| 15 | **PmeStatus: PME status**. Read-only. |
| 14:13 | **DataScale: data scale**. Read-only. |
| 12:9 | **DataSelect: data select**. Read-only. |
| 8 | **PmeEn: PME# enable**. Read-only. |
| 7:4 | Reserved. |
| 3 | **NoSoftReset: no soft reset**. Read-only. Software is required to re-initialize the function when returning from $D3_{hot}$. |
| 2 | Reserved. |
| 1:0 | **PowerState: power state**. Read-write. This 2-bit field is used both to determine the current power state of the root port and to set the root port into a new power state.<br>Bits Definition<br>00b D0<br>11b D3hot |

### D1F1x58 PCI Express Capability

| Bits | Description |
|---|---|
| 31:30 | Reserved. |
| 29:25 | **IntMessageNum: interrupt message number**. Value: 0. This field indicates which MSI vector is used for the interrupt message. |

| | |
|---|---|
| 24 | **SlotImplemented: Slot implemented**. Value: 0. |
| 23:20 | **DeviceType: device type**. Value: IF (D0F0x64_x1C[AudioNonlegacyDeviceTypeEn]==0) THEN 1. ELSE 0. ENDIF. |
| 19:16 | **Version**. Value: 2h. |
| 15:8 | **NextPtr: next pointer**. Value: IF (D0F0x64_x1C[MsiDis]==0) THEN A0h. ELSE 00h. ENDIF. |
| 7:0 | **CapID: capability ID**. Value: 10h. |

### D1F1x5C Device Capability

| Bits | Description |
|---|---|
| 31:29 | Reserved. |
| 28 | **FlrCapable: function level reset capability**. Value: 0. |
| 27:26 | **CapturedSlotPowerScale: captured slot power limit scale**. Value: 0. |
| 25:18 | **CapturedSlotPowerLimit: captured slot power limit value**. Value: 0. |
| 17:16 | Reserved. |
| 15 | **RoleBasedErrReporting: role-based error reporting**. Value: 1. |
| 14:12 | Reserved. |
| 11:9 | **L1AcceptableLatency: endpoint L1 Acceptable Latency**. Value: 111b. |
| 8:6 | **L0SAcceptableLatency: endpoint L0s Acceptable Latency**. Value: 110b. |
| 5 | **ExtendedTag: extended tag support**. Value: 1. 8 bit tag support. |
| 4:3 | **PhantomFunc: phantom function support**. Value: 0. No phantom functions supported. |
| 2:0 | **MaxPayloadSupport: maximum supported payload size**. Value: 000b. 128 bytes max payload size. |

### D1F1x60 Device Control and Status

Reset: 0000_0810h.

| Bits | Description |
|---|---|
| 31:22 | Reserved. |
| 21 | **TransactionsPending: transactions pending**. Read-only. |
| 20 | **AuxPwr: auxiliary power**. Read-only. |
| 19 | **UsrDetected: unsupported request detected**. Read; Write-1-to-clear. 1=Unsupported request received. |
| 18 | **FatalErr: fatal error detected**. Read; Write-1-to-clear. 1=Fatal error detected. |
| 17 | **NonFatalErr: non-fatal error detected**. Read; Write-1-to-clear. 1=Non-fatal error detected. |
| 16 | **CorrErr: correctable error detected**. Read; Write-1-to-clear. 1=Correctable error detected. |
| 15 | **BridgeCfgRetryEn: bridge configuration retry enable**. Read-only. |
| 14:12 | **MaxRequestSize: maximum request size**. Read-only. 0=The root port never generates read requests with size exceeding 128 bytes. |

| | |
|---|---|
| 11 | **NoSnoopEnable: enable no snoop**. Read-write. 1=the device is permitted to set the No Snoop bit in requests. |
| 10 | **AuxPowerPmEn: auxiliary power PM enable**. Read-only. This capability is not implemented. |
| 9 | **PhantomFuncEn: phantom functions enable**. Read-only. Phantom functions are not supported. |
| 8 | **ExtendedTagEn: extended tag enable**. Read-write. 1=8-bit tag request tags. 0=5-bit request tag. |
| 7:5 | **MaxPayloadSize: maximum supported payload size**. Read-only. 000b=Indicates a 128 byte maximum payload size. |
| 4 | **RelaxedOrdEn: relaxed ordering enable**. Read-write. 1=The device is permitted to set the Relaxed Ordering bit. |
| 3 | **UsrReportEn: unsupported request reporting enable**. Read-write. 1=Enables signaling unsupported requests by sending error messages. |
| 2 | **FatalErrEn: fatal error reporting enable**. Read-write. 1=Enables sending ERR_FATAL message when a fatal error is detected. |
| 1 | **NonFatalErrEn: non-fatal error reporting enable**. Read-write. 1=Enables sending ERR_NONFATAL message when a non-fatal error is detected. |
| 0 | **CorrErrEn: correctable error reporting enable**. Read-write. 1=Enables sending ERR_CORR message when a correctable error is detected. |

### D1F1x64 Link Capability

| Bits | Description |
|---|---|
| 31:24 | **PortNumber: port number**. Value: 0.This field indicates the PCI Express port number for the given PCI Express link. |
| 23:22 | Reserved. |
| 21 | **LinkBWNotificationCap: link bandwidth notification capability**. Value: 0b. |
| 20 | **DlActiveReportingCapable: data link layer active reporting capability**. Value: 0b. |
| 19 | **SurpriseDownErrReporting: surprise down error reporting capability**. Value: 0b. |
| 18 | **ClockPowerManagement: clock power management**. Value: 0b. |
| 17:15 | **L1ExitLatency: L1 exit latency**.Value: 000b. |
| 14:12 | **L0sExitLatency: L0s exit latency**. Value: 000b. |
| 11:10 | **PMSupport: active state power management support**. Value: 00b. |
| 9:4 | **LinkWidth: maximum link width**. Value: 0. |
| 3:0 | **LinkSpeed: link speed**.Value: 0. |

### D1F1x68 Link Control and Status

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31 | **LinkAutonomousBWStatus: link autonomous bandwidth status**. Read-only. |
| 30 | **LinkBWManagementStatus: link bandwidth management status**. Read-only. |

| | |
|---|---|
| 29 | **DlActive: data link layer link active**. Read-only. This bit indicates the status of the data link control and management state machine. Reads return a 1 to indicate the DL_Active state, otherwise 0 is returned. |
| 28 | **SlotClockCfg: slot clock configuration**. Read-only. 1=the root port uses the same clock that the platform provides. |
| 27 | **LinkTraining: link training**. Read-only. 1=Indicates that the physical layer link training state machine is in the configuration or recovery state, or that 1b was written to the RetrainLink bit but link training has not yet begun. Hardware clears this bit when the link training state machine exits the configuration/recovery state. |
| 26 | Reserved. |
| 25:20 | **NegotiatedLinkWidth: negotiated link width**. Read-only. This field indicates the negotiated width of the given PCI Express link. |
| 19:16 | **LinkSpeed: link speed**. Read-only.<br>0001b: 2.5 Gb/s.<br>0010b: 5 Gb/s |
| 15:12 | Reserved. |
| 11 | **LinkAutonomousBWIntEn: link autonomous bandwidth interrupt enable**. Read-only. |
| 10 | **LinkBWManagementEn: link bandwidth management interrupt enable**. Read-only. |
| 9 | **HWAutonomousWidthDisable: hardware autonomous width disable**. Read-write. 1=Hardware not allowed to change the link width except to correct unreliable link operation by reducing link width. |
| 8 | **ClockPowerManagementEn: clock power management enable**. Read-write. |
| 7 | **ExtendedSync: extended sync**. Read-write. 1=Forces the transmission of additional ordered sets when exiting the L0s state and when in the recovery state. |
| 6 | **CommonClockCfg: common clock configuration**. Read-write. 1=Indicates that the root port and the component at the opposite end of this Link are operating with a distributed common reference clock. 0=Indicates that the upstream port and the component at the opposite end of this Link are operating with asynchronous reference clock. |
| 5 | **RetrainLink: retrain link**. Read-only. This bit does not apply to endpoints. |
| 4 | **LinkDis: link disable**. Read-only. This bit does not apply to endpoints. |
| 3 | **ReadCplBoundary: read completion boundary**. Read-only. 0=64 byte read completion boundary. |
| 2 | Reserved. |
| 1:0 | **PmControl: active state power management enable**. Read-write. This field controls the level of ASPM supported on the given PCI Express link.<br>Bits / Definition / Bits / Definition<br>00b Disabled. 10b L1 Entry Enabled.<br>01b L0s Entry Enabled. 11b L0s and L1 Entry Enabled. |

### D1F1x7C Device Capability 2

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:5 | Reserved. |

| | |
|---|---|
| 4 | **CplTimeoutDisSup: completion timeout disable supported**. Read-only. |
| 3:0 | **CplTimeoutRangeSup: completion timeout range supported**. Read-only. |

### D1F1x80 Device Control and Status 2

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:5 | Reserved. |
| 4 | **CplTimeoutDis: completion timeout disable**. Read-only. |
| 3:0 | **CplTimeoutValue: completion timeout range supported**. Read-only. |

### D1F1x84 Link Capability 2

| Bits | Description |
|---|---|
| 31:0 | Reserved. |

### D1F1x88 Link Control and Status 2

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:17 | Reserved. |
| 16 | **CurDeemphasisLevel: current deemphasis level**. Read-only.<br>1: -3.5 dB<br>0: -6 dB |
| 15:13 | Reserved. |
| 12 | **ComplianceDeemphasis: compliance deemphasis**. Read-write. This bit defines the deemphasis level used in compliance mode.<br>1: -3.5 dB<br>0: -6 dB |
| 11 | **ComplianceSOS: compliance SOS**. Read-write. 1=The device transmits skip ordered sets in between the modified compliance pattern. |
| 10 | **EnterModCompliance: enter modified compliance**. Read-write. 1=The device transmits modified compliance pattern. |
| 9:7 | **XmitMargin: transmit margin**. Read-write. This field controls the non-deemphasized voltage level at the transmitter pins. |
| 6 | **SelectableDeemphasis: selectable deemphasis**. Read-only. |
| 5 | **HwAutonomousSpeedDisable: hardware autonomous speed disable**. Read-write. 1=Disables hardware generated link speed changes. |
| 4 | **EnterCompliance: enter compliance**. Read-write. 1=Force link to enter compliance mode. |
| 3:0 | **TargetLinkSpeed: target link speed**. Read-write. This fields defines the upper limit of the link operational speed. |

### D1F1xA0 MSI Capability

| Bits | Description |
|---|---|
| 31:24 | Reserved. |
| 23 | **Msi64bit: MSI 64 bit capability**. Read-only. Value: D0F0x64_x46[Msi64bitEn]. 1=The device is capable of sending 64-bit MSI messages. 0=The device is not capable of sending a 64-bit message address. |
| 22:20 | **MsiMultiEn: MSI multiple message enable**. Read-write. Reset: 000b. Software writes to this field to indicate the number of allocated vectors (equal to or less than the number of requested vectors). When MSI is enabled, a function is allocated at least 1 vector. |
| 19:17 | **MsiMultiCap: MSI multiple message capability**. Read-only. Reset:000b. 000b=The device is requesting one vector. |
| 16 | **MsiEn: MSI enable**. Read-write. Reset: 0. 1=MSI generation is enabled and INTx generation is disabled. 0=MSI generation disabled and INTx generation is enabled. |
| 15:8 | **NextPtr: next pointer**. Read-only. Reset:00h |
| 7:0 | **CapID: capability ID**. Read-only. Reset: 05h. 05h=MSI capability structure. |

### D1F1xA4 MSI Message Address Low

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:2 | **MsiMsgAddrLo: MSI message address**. Read-write. This register specifies the dword aligned address for the MSI memory write transaction. |
| 1:0 | Reserved. |

IF (D0F0x64_x46[Msi64bitEn]==1) THEN

### D1F1xA8 MSI Message Address High

ENDIF.
IF (D0F0x64_x46[Msi64bitEn]==0) THEN

### D1F1xA8 MSI Message Data

ELSEIF (D0F0x64_x46[Msi64bitEn]==1) THEN

### D1F1xAC MSI Message Data

ENDIF.
Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:16 | Reserved. |
| 15:0 | **MsiData: MSI message data**. Read-write. This register specifies lower 16 bits of data for the MSI memory write transaction. The upper 16 bits are always 0. |

**D1F1x100 Vendor Specific Enhanced Capability**

Reset: 0001_000Bh.

| Bits | Description |
|---|---|
| 31:20 | **NextPtr: next pointer**. Read-only. |
| 19:16 | **CapVer: capability version**. Read-only. |
| 15:0 | **CapID: capability ID**. Read-only. |

**D1F1x104 Vendor Specific Header**

Reset: 0101_0001h.

| Bits | Description |
|---|---|
| 31:20 | **VsecLen: vendor specific enhanced next pointer**. Read-only. |
| 19:16 | **VsecRev: vendor specific enhanced capability version**. Read-only. |
| 15:0 | **VsecID: vendor specific enhanced capability ID**. Read-only. |

**D1F1x108 Vendor Specific 1**

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:0 | **Scratch: scratch**. Read-write. |

**D1F1x10C Vendor Specific 2**

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:0 | **Scratch: scratch**. Read-write. |

## 3.6 Device [8:2] Function 0 (Root Port) Configuration Registers

See 3.1 [Register Descriptions and Mnemonics] for a description of the register naming convention. See 2.7 [Configuration Space] for details about how to access this space.

**D[8:2]F0x00 Device/Vendor ID Register**

**Table 71: Reset mapping for** D[8:2]F0x00**.**

| Register | Reset | Register | Reset |
|---|---|---|---|
| D2F0x00 | 1707_1022h. | D6F0x00 | 170B_1022h. |
| D3F0x00 | 1708_1022h. | D7F0x00 | 170C_1022h. |
| D4F0x00 | 1709_1022h. | D8F0x00 | 170D_1022h. |
| D5F0x00 | 170A_1022h. | - | - |

| Bits | Description |
|---|---|
| 31:16 | **DeviceID: device ID**. Read-only. |
| 15:0 | **VendorID: vendor ID**. Read-only. |

**D[8:2]F0x04 Status/Command Register**

Reset: 0010_0000h.

| Bits | Description |
|---|---|
| 31 | **ParityErrorDetected: detected parity error**. Read; Write-1-to-clear. |
| 30 | **SignaledSystemError: signaled system error**. Read; Write-1-to-clear. 1=System error signaled. |
| 29 | **ReceivedMasterAbort: received master abort**. Read; Write-1-to-clear. |
| 28 | **ReceivedTargetAbort: received target abort**. Read; Write-1-to-clear. |
| 27 | **SignaledTargetAbort: signaled target abort**. Read; Write-1-to-clear. |
| 26:25 | **DevselTiming: DEVSEL# Timing**. Read-only. |
| 24 | **MasterDataPerr: master data parity error**. Read; Write-1-to-clear. |
| 23 | **FastBackCapable: fast back-to-back capable**. Read-only. |
| 22 | Reserved. |
| 21 | **PCI66En: 66 MHz capable**. Read-only. |
| 20 | **CapList: capability list**. Read-only. 1= Capability list present. |
| 19 | **IntStatus: interrupt status**. Read-only. 1=An INTx interrupt Message is pending in the device. |
| 18:11 | Reserved. |
| 10 | **IntDis: interrupt disable**. Read-write. |
| 9 | **FastB2BEn: fast back-to-back enable**. Read-only. |
| 8 | **SerrEn: system error enable**. Read-write. 1=System error reporting enabled. |
| 7 | **IdselStepping: IDSEL stepping control**. Read-only. |
| 6 | **ParityErrorEn: parity error response enable**. Read-write. |
| 5 | **PalSnoopEn: VGA palette snoop enable**. Read-only. |
| 4 | **MemWriteInvalidateEn: memory write and invalidate enable**. Read-only. |
| 3 | **SpecialCycleEn: special cycle enable**. Read-only. |
| 2 | **BusMasterEn: bus master enable**. Read-write. |
| 1 | **MemAccessEn: memory access enable**. Read-write. This bit controls if memory accesses targeting this device are accepted or not. 1=Enabled. 0=Disabled. |
| 0 | **IoAccessEn: IO access enable**. Read-write. This bit controls if IO accesses targeting this device are accepted or not. 1=Enabled. 0=Disabled. |

### D[8:2]F0x08 Class Code/Revision ID Register

Reset: 0604_00xxh.

| Bits | Description |
|---|---|
| 31:8 | **ClassCode**. Read-only. Provides the host bridge class code as defined in the PCI specification. |
| 7:0 | **RevID: revision ID**. Read-only. |

### D[8:2]F0x0C Header Type Register

Reset: 0001_0000h.

| Bits | Description |
|---|---|
| 31:24 | **BIST**. Read-only. These bits are fixed at their default values. |
| 23:16 | **HeaderTypeReg**. Read-only. These bits are fixed at their default values. The header type field indicates a header type 1 and that there is only one function present in this device. |
| 15:8 | **LatencyTimer**. Read-only. This field does not control any hardware. |
| 7:0 | **CacheLineSize**. Read-write. |

### D[8:2]F0x18 Bus Number and Secondary Latency Register

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:24 | **SecondaryLatencyTimer: secondary latency timer**. Read-only. This field is always 0. |
| 23:16 | **SubBusNumber: subordinate number**. Read-write. This field contains the highest-numbered bus that exists on the secondary side of the bridge. |
| 15:8 | **SecondaryBus: secondary bus number**. Read-write. This field defines the bus number of the secondary bus interface. |
| 7:0 | **PrimaryBus: primary bus number**. Read-write. This field defines the bus number of the primary bus interface. |

### D[8:2]F0x1C IO Base and Secondary Status Register

Reset: 0000_0101h.

| Bits | Description |
|---|---|
| 31 | **ParityErrorDetected: detected parity error**. Read; Write-1-to-clear. A Poisoned TLP was received regardless of the state of the D[8:2]F0x04[ParityErrorEn]. |
| 30 | **ReceivedSystemError: signaled system error**. Read; Write-1-to-clear. 1=A System Error was detected. |
| 29 | **ReceivedMasterAbort: received master abort**. Read; Write-1-to-clear. 1=A CPU transaction is terminated due to a master-abort. |
| 28 | **ReceivedTargetAbort: received target abort**. Read; Write-1-to-clear. 1=A CPU transaction (except for a special cycle) is terminated due to a target-abort. |
| 27 | **SignalTargetAbort: signaled target abort**. Read; Write-1-to-clear. |

| | |
|---|---|
| 26:25 | **DevselTiming: DEVSEL# Timing**. Read-only. |
| 24 | **MasterDataPerr: master data parity error**. Read; Write-1-to-clear. 1=The link received a poisoned or poisoned a downstream write and D[8:2]F0x3C[ParityResponseEn]=1. |
| 23 | **FastBackCapable: fast back-to-back capable**. Read-only. |
| 22 | Reserved. |
| 21 | **PCI66En: 66 MHz capable**. Read-only. |
| 20:16 | Reserved. |
| 15:12 | **IOLimit[15:12]**. Read-write. Lower part of the limit address. Upper part is defined in D[8:2]F0x30. |
| 11:8 | Reserved. |
| 7:4 | **IOBase[15:12]**. Read-write. Lower part of the base address. Upper part is defined in D[8:2]F0x30. |
| 3:0 | Reserved. |

### D[8:2]F0x20 Memory Limit and Base Register

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:20 | **MemLimit**. Read-write. |
| 19:16 | Reserved. |
| 15:4 | **MemBase**. Read-write. |
| 3:0 | Reserved. |

### D[8:2]F0x24 Prefetchable Memory Limit and Base Register

Reset: 0001_0001h.

| Bits | Description |
|---|---|
| 31:20 | **PrefMemLimit**. Read-write. Lower part of the limit address. Upper part is defined in D[8:2]F0x2C. |
| 19:16 | **PrefMemLimitR**. Read-only. 1=64 bit memory address decoder. |
| 15:4 | **PrefMemBase[31:20]**. Read-write. Lower part of the base address. Upper part is defined in D[8:2]F0x28. |
| 3:0 | **PrefMemBaseR**. Read-only. 1= 64 bit memory address decoder. |

### D[8:2]F0x28 Prefetchable Memory Base High Register

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:0 | **PrefMemBase[63:32]**. Read-write. Upper part of the base address. Lower part is defined in D[8:2]F0x24. |

### D[8:2]F0x2C Prefetchable Memory Limit High Register

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:0 | **PrefMemLimit[63:32]**. Read-write. Upper part of the limit address. Lower part is defined in D[8:2]F0x24. |

### D[8:2]F0x30 IO Base and Limit High Register

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:16 | **IOLimit[31:16]**. Read-write. Upper part of the limit address. Lower part is defined in D[8:2]F0x1C. |
| 15:0 | **IOBase[31:16]**. Read-write. Upper part of the base address. Lower part is defined in D[8:2]F0x1C. |

### D[8:2]F0x34 Capabilities Pointer Register

Reset: 0000_0050h.

| Bits | Description |
|---|---|
| 31:8 | Reserved. |
| 7:0 | **CapPtr: capabilities pointer**. Read-only. Pointer to PM capability. |

### D[8:2]F0x3C Bridge Control Register

Reset: 0000_00FFh.

| Bits | Description |
|---|---|
| 31:24 | Reserved. |
| 23 | **FastB2BCap: Fast back-to-back capability**. Read-only. |
| 22 | **SecondaryBusReset: Secondary bus reset**. Read-write. Setting this bit triggers a hot reset on the corresponding PCI Express Port. |
| 21 | **MasterAbortMode: Master abort mode**. Read-only. |
| 20 | **Vga16En: VGA IO 16 bit decoding enable**. Read-write. 1= Address bits 15:10 for VGA IO cycles are decoded. 0=Address bits 15:10 for VGA IO cycles are ignored. |
| 19 | **VgaEn: VGA enable**. Read-write. Affects the response by the bridge to compatible VGA addresses. When it is set, the bridge decodes and forwards the following accesses on the primary interface to the secondary interface:<br>Memory accesses in the range of A_0000h to B_FFFFh and<br>IO address where address bits 9:0 are in the ranges of 3B0h to 3BBh or 3C0h to 3DFh. For IO cycles the decoding of address bits 15:10 depends on Vga16En. |
| 18 | **IsaEn: ISA enable**. Read-write. |

| | |
|---|---|
| 17 | **SerrEn: SERR enable**. Read-write. |
| 16 | **ParityResponseEn: Parity response enable**. Read-write. Controls the bridge's response to poisoned TLPs on its secondary interface. 1=The bridge takes its normal action when a poisoned TLP is received. 0=The bridge ignores any poisoned TLPs that it receives and continues normal operation. |
| 15:11 | **IntPinR: interrupt pin**. Read-only. |
| 10:8 | **IntPin: interrupt pin**. IF (D0F0xE4_x0[2:1]01_0010[HwInitWrLock]==1) THEN Read-only. ELSE Read-write. ENDIF. |
| 7:0 | **IntLine: Interrupt line**. Read-write. |

### D[8:2]F0x50 Power Management Capability Register

Reset: C803_5801h.

| Bits | Description |
|---|---|
| 31:27 | **PmeSupport**. Read-only. |
| 26 | **D2Support: D2 support**. Read-only. D2 is not supported |
| 25 | **D1Support: D1 support**. Read-only. D1 is not supported |
| 24:22 | **AuxCurrent: auxiliary current**. Read-only. Auxiliary current is not supported. |
| 21 | **DevSpecificInit: device specific initialization**. Read-only. This field is hardwired to 0 to indicate that there is no device specific initialization necessary. |
| 20 | Reserved. |
| 19 | **PmeClock**. Read-only. 0=Indicate that PCI clock is not needed to generate PME messages. |
| 18:16 | **Version: version**. Read-only. |
| 15:8 | **NextPtr: next pointer**. Read-only. 58h=Address of the next capability structure. |
| 7:0 | **CapID: capability ID**. Read-only. 01h=PCI power management data structure. |

### D[8:2]F0x54 Power Management Control and Status Register

| Bits | Description |
|---|---|
| 31:24 | **PmeData**. Read-only. Reset: 0. |
| 23 | **BusPwrEn**. Read-only. Reset: 0. |
| 22 | **B2B3Support**. Read-only. Reset: 0. B states are not supported. |
| 21:16 | Reserved. |
| 15 | **PmeStatus: PME status**. Read; Write-1-to-clear. Reset: 0. This bit is set when the root port would issue a PME message (independent of the state of the PmeEn bit). Once set, this bit remains set until it is reset by writing a 1 to this bit location. Writing a 0 has no effect. |
| 14:13 | **DataScale: data scale**. Read-only. Reset: 0. |
| 12:9 | **DataSelect: data select**. Read-only. Reset: 0. |
| 8 | **PmeEn: PME# enable**. Read-write. Reset: 0. |
| 7:4 | Reserved. |
| 3 | **NoSoftReset: no soft reset**. Read-only. Reset: 0. Software is required to re-initialize the function when returning from $D3_{hot}$. |

| | |
|---|---|
| 2 | Reserved. |
| 1:0 | **PowerState: power state**. Read-write. Reset: 0. This 2-bit field is used both to determine the current power state of the root port and to set the root port into a new power state.<br>Bits Definition Bits Definition<br>00b D0 10b Reserved<br>01b Reserved 11b D3 |

**D[8:2]F0x58 PCI Express Capability Register**

Reset: 0042_A010h.

| Bits | Description |
|---|---|
| 31:30 | Reserved. |
| 29:25 | **IntMessageNum: interrupt message number**. Read-only. This register indicates which MSI vector is used for the interrupt message. |
| 24 | **SlotImplemented: Slot implemented**. Read-only. 1=The IO Link associated with this port is connected to a slot. |
| 23:20 | **DeviceType: device type**. Read-only. 4h=Root complex. |
| 19:16 | **Version**. Read-only. 2h=GEN 2 compliant. |
| 15:8 | **NextPtr: next pointer**. Read-only. A0h=Pointer to the next capability structure. |
| 7:0 | **CapID: capability ID**. Read-only. 10h=PCIe® Capability structure. |

**D[8:2]F0x5C Device Capability Register**

Reset: 0000_8020h.

| Bits | Description |
|---|---|
| 31:29 | Reserved. |
| 28 | **FlrCapable: function level reset capability**. Read-only. |
| 27:26 | **CapturedSlotPowerScale: captured slot power limit scale**. Read-only. |
| 25:18 | **CapturedSlotPowerLimit: captured slot power limit value**. Read-only. |
| 17:16 | Reserved. |
| 15 | **RoleBasedErrReporting: role-based error reporting**. Read-only. |
| 14:12 | Reserved. |
| 11:9 | **L1AcceptableLatency: endpoint L1 Acceptable Latency**. Read-only. |
| 8:6 | **L0SAcceptableLatency: endpoint L0s Acceptable Latency**. Read-only. |
| 5 | **ExtendedTag: extended tag support**. Read-only.<br>1: 8 bit tag supported<br>0: 5 bit tag supported. |
| 4:3 | **PhantomFunc: phantom function support**. Read-only. 0=No phantom functions supported. |
| 2:0 | **MaxPayloadSupport: maximum supported payload size**. Read-only. 000b=128 bytes max payload size. |

**D[8:2]F0x60 Device Control and Status Register**

Reset: 0000_2810h.

| Bits | Description |
|---|---|
| 31:22 | Reserved. |
| 21 | **TransactionsPending: transactions pending**. Read-only. 0=No internally generated non-posted transactions pending. |
| 20 | **AuxPwr: auxiliary power**. Read-only. |
| 19 | **UsrDetected: unsupported request detected**. Read; Write-1-to-clear. 1=The port received an unsupported request. Errors are logged in this register even if error reporting is disabled. |
| 18 | **FatalErr: fatal error detected**. Read; Write-1-to-clear. 1=The port detected a fatal error. Errors are logged in this register even if error reporting is disabled. |
| 17 | **NonFatalErr: non-fatal error detected**. Read; Write-1-to-clear. T1=The port detected a non-fatal error. Errors are logged in this register even if error reporting is disabled. |
| 16 | **CorrErr: correctable error detected**. Read; Write-1-to-clear. 1=The port detected a correctable error. Errors are logged in this register even if error reporting is disabled. |
| 15 | **BridgeCfgRetryEn: bridge configuration retry enable**. Read-only. |
| 14:12 | **MaxRequestSize: maximum request size**. Read-write. |
| 11 | **NoSnoopEnable: enable no snoop**. Read-write. 1=The port is permitted to set the No Snoop bit in the Requester Attributes of transactions it initiates that do not require hardware enforced cache coherency. |
| 10 | **AuxPowerPmEn: auxiliary power PM enable**. Read-only. |
| 9 | **PhantomFuncEn: phantom functions enable**. Read-only. |
| 8 | **ExtendedTagEn: extended tag enable**. Read-write. 1=8-bit tags generation enabled. 0=5-bit tags are used. |
| 7:5 | **MaxPayloadSize: maximum supported payload size**. Read-write.<br>Bits Definition Bits Definition<br>0h 128B 3h 1024B<br>1h 256B 4h 2048B<br>2h 512B 5h 4096B |
| 4 | **RelaxedOrdEn: relaxed ordering enable**. Read-write. 1=The root port is permitted to set the relaxed ordering bit in the attributes field of transactions it initiates that do not require strong write ordering. |
| 3 | **UsrReportEn: unsupported request reporting enable**. Read-write. 1=Reporting of unsupported requests enabled. |
| 2 | **FatalErrEn: fatal error reporting enable**. Read-write. 1=Enable sending ERR_FATAL messages. |
| 1 | **NonFatalErrEn: non-fatal error reporting enable**. Read-write. 1=Enable sending ERR_NONFATAL messages. |
| 0 | **CorrErrEn: correctable error reporting enable**. Read-write. 1=Enable sending ERR_CORR messages. |

### D[8:2]F0x64 IO Link Capability Register

Read-only. Reset: F710_0D02h.

| Bits | Description |
|---|---|
| 31:24 | **PortNumber: port number**. This field indicates the port number for the given IO link. |
| 23:22 | Reserved. |
| 21 | **LinkBWNotificationCap: link bandwidth notification capability**. |
| 20 | **DlActiveReportingCapable: data link layer active reporting capability**. |
| 19 | Reserved. |
| 18 | **ClockPowerManagement: clock power management**. 0=Indicates that the reference clock must not be removed while in L1 or L2/L3 ready link states. |
| 17:15 | **L1ExitLatency: L1 exit latency**. 010b=Indicate an exit latency between 2 us and 4 us. |
| 14:12 | **L0sExitLatency: L0s exit latency**. 001b=Indicates an exit latency between 64 ns and 128 ns. |
| 11:10 | **PMSupport: active state power management support**.11b=Indicates support of L0s and L1. |
| 9:4 | **LinkWidth: maximum link width**.<br>Bits — Definition<br>01h — 1 lanes<br>02h — 2 lanes<br>04h — 4 lanes<br>08h — 8 lanes<br>0Ch — 12 lanes<br>10h — 16 lanes |
| 3:0 | **LinkSpeed: link speed**.<br>Bits — Definition<br>01h — 2.5 Gb/s<br>02h — 5.0 Gb/s |

### D[8:2]F0x68 IO Link Control and Status Register

Reset: 1001_0000h.

| Bits | Description |
|---|---|
| 31 | **LinkAutonomousBWStatus: link autonomous bandwidth status**. IF (D[8:2]F0x64[LinkBWNotificationCap]==0) THEN Read-only. ELSE Read-write. ENDIF. |
| 30 | **LinkBWManagementStatus: link bandwidth management status**. IF (D[8:2]F0x64[LinkBWNotificationCap]==0) THEN Read-only. ELSE Read-write. ENDIF. |
| 29 | **DlActive: data link layer link active**. Read-only. This bit indicates the status of the data link control and management state machine. 1=DL_Active state. 0=All other states. |
| 28 | **SlotClockCfg: slot clock configuration**. Read-only. 1=The root port uses the same clock that the platform provides. |
| 27 | **LinkTraining: link training**. Read-only. This read-only bit indicates that the physical layer link training state machine is in the configuration or recovery state, or that 1b was written to the RetrainLink bit but link training has not yet begun. Hardware clears this bit when the link training state machine exits the configuration/recovery state. |

| | |
|---|---|
| 26 | Reserved. |
| 25:20 | **NegotiatedLinkWidth: negotiated link width**. Read-only. This field indicates the negotiated width of the given PCI Express link.<br>Bits — Definition<br>01h — 1 lanes<br>02h — 2 lanes<br>04h — 4 lanes<br>08h — 8 lanes<br>0Ch — 12 lanes<br>10h — 16 lanes |
| 19:16 | **LinkSpeed: link speed**. Read-only.<br>Bits — Definition<br>0001b — 2.5 Gb/s<br>0010b — 5 Gb/s |
| 15:12 | Reserved. |
| 11 | **LinkAutonomousBWIntEn: link autonomous bandwidth interrupt enable**. Read-write. 1=Enables the generation of an interrupt to indicate that the Link AutonomousBWStatus bit has been set. |
| 10 | **LinkBWManagementEn: link bandwidth management interrupt enable**. Read-write. 1=Enables the generation of an interrupt to indicate that the LinkBWManagementStatus has been set. |
| 9 | **HWAutonomousWidthDisable: hardware autonomous width disable**. Read-write. 1=Disables hardware from changing the link width for reasons other than attempting to correct unreliable link operation by reducing link width. |
| 8 | **ClockPowerManagementEn: clock power management enable**. Read-write. |
| 7 | **ExtendedSync: extended sync**. Read-write. 1=Forces the transmission of additional ordered sets when exiting the L0s state and when in the recovery state. |
| 6 | **CommonClockCfg: common clock configuration**. Read-write. 1=Indicates that the root port and the component at the opposite end of this IO link are operating with a distributed common reference clock. 0=Indicates that the root port and the component at the opposite end of this IO Link are operating with asynchronous reference clock. |
| 5 | **RetrainLink: retrain link**. Read-write; cleared-when-done. 1=Initiate link retraining. |
| 4 | **LinkDis: link disable**. Read-write. 1=Disable link. Writes to this bit are immediately reflected in the value read from the bit, regardless of actual link state. |
| 3 | **ReadCplBoundary: read completion boundary**. Read-only. 0=64 byte read completion boundary. |
| 2 | Reserved. |
| 1:0 | **PmControl: active state power management enable**. Read-write. This field controls the level of ASPM supported on the given IO link.<br>Bits — Definition — Bits — Definition<br>00b — Disabled. — 10b — L1 Entry Enabled.<br>01b — L0s Entry Enabled. — 11b — L0s and L1 Entry Enabled. |

### D[8:2]F0x6C Slot Capability Register

Reset: 0004_0000h.

| Bits | Description |
|---|---|
| 31:19 | **PhysicalSlotNumber: physical slot number**. Read-write.<br>This field indicates the physical slot number attached to this port. This field is set to a value that assigns a slot number that is unique within the chassis, regardless of the form factor associated with the slot. This field must be initialized to 0 for ports connected to devices that are on the system board. |
| 18 | **NoCmdCplSupport: no command completed support**. Read-write. 1 =Indicates that this slot does not generate software notification when an issued command is completed by the hot-plug controller. |
| 17 | **ElecMechIlPresent: electromechanical interlock present**. Read-write. 0=Indicates that a electro-mechanical interlock is not implemented for this slot. |
| 16:15 | **SlotPwrLimitScale: slot power limit scale**. Read-write. Specifies the scale used for the SlotPwrLimitValue. Range of Values:<br>Bits Definition Bits Definition<br>00b 1.0 10b 0.01<br>01b 0.1 11b 0.001 |
| 14:7 | **SlotPwrLimitValue: slot power limit value**. Read-write. In combination with the SlotPwrLimitScale value, specifies the upper limit on power supplied by slot. Power limit (in Watts) calculated by multiplying the value in this field by the value in the SlotPwrLimitScale field. |
| 6 | **HotplugCapable: hot-plug capability**. Read-write.1=Indicates that this slot is capable of supporting hot-plug operations. |
| 5 | **HotplugSurprise: hot-plug surprise**. Read-write. 1=Indicates that an adapter present in this slot might be removed from the system without any prior notification. |
| 4 | **PwrIndicatorPresent: power indicator present**. Read-write. 0=Indicates that a power indicator is not implemented for this slot. |
| 3 | **AttnIndicatorPresent: attention indicator present**. Read-write. 0=Indicates that a attention indicator is not implemented for this slot. |
| 2 | **MrlSensorPresent: manual retention latch sensor present**. Read-write. 0=Indicates that a manual retention latch sensor is not implemented for this slot. |
| 1 | **PwrControllerPresent: power controller present**. Read-write. 0=A power controller is not implemented for this slot. |
| 0 | **AttnButtonPresent: attention button present**. Read-write. 0=An attention button is not implemented for this slot. |

### D[8:2]F0x70 Slot Control and Status Register

IF (D[8:2]F0x58[SlotImplemented]==0) THEN Reset: 0040_0000h. ELSE Reset: 0000_0000h. ENDIF.

| Bits | Description |
|---|---|
| 31:25 | Reserved. |
| 24 | **DlStateChanged: data link layer state change**. Read; Write-1-to-clear. This bit is set when the value reported in the D[8:2]F0x60[DlActive] is changed. In response to a data link layer state changed event, software must read D[8:2]F0x60[DlActive] to determine if the link is active before initiating configuration cycles to the hot plugged device. |

| | |
|---|---|
| 23 | **ElecMechIlSts: electromechanical interlock status**. Read-only. |
| 22 | **PresenceDetectState: presence detect state**. Read-only. This bit indicates the presence of an adapter in the slot based on the physical layer in-band presence detect mechanism. The in-band presence detect mechanism requires that power be applied to an adapter for its presence to be detected.<br>0=Slot empty.<br>1=Card present in slot.<br>For root ports not connected to slots (D[8:2]F0x58[SlotImplemented]=0b), this bit returns always 1. |
| 21 | **MrlSensorState**. Read-only. |
| 20 | **CmdCpl: command completed**. Read-only. |
| 19 | **PresenceDetectChanged: presence detect changes**. Read; Write-1-to-clear. This bit is set when the value reported in PresenceDetectState is changed. |
| 18 | **MrlSensorChanged**. Read; Write-1-to-clear. |
| 17 | **PwrFaultDetected**. Read; Write-1-to-clear. |
| 16 | **AttnButtonPressed: attention button pressed**. Read-only. |
| 15:13 | Reserved. |
| 12 | **DlStateChangedEn: data link layer state changed enable**. Read-write. 1=Enables software notification when D[8:2]F0x60[DlActive] is changed. |
| 11 | **ElecMechIlCntl: electromechanical interlock control**. Read-only. |
| 10 | **PwrControllerCntl: power controller control**. Read-only. |
| 9:8 | **PwrIndicatorCntl: power indicator control**. Read-only. |
| 7:6 | **AttnIndicatorControl: attention indicator control**. Read-only. |
| 5 | **HotplugIntrEn: hot-plug interrupt enable**. Read-only. |
| 4 | **CmdCplIntrEn: command complete interrupt enable**. Read-only. |
| 3 | **PresenceDetectChangedEn: presence detect changed enable**. Read-only. |
| 2 | **MrlSensorChangedEn: manual retention latch sensor changed enable**. Read-only. |
| 1 | **PwrFaultDetectedEn: power fault detected enable**. Read-only. |
| 0 | **AttnButtonPressedEn: attention button pressed enable**. Read-only. |

### D[8:2]F0x74 Root Complex Capability and Control Register

Reset: 0001_0000h.

| Bits | Description |
|---|---|
| 31:17 | Reserved. |
| 16 | **CrsSoftVisibility: CRS software visibility**. Read-only. 1=Indicates that the root port supports returning configuration request retry status (CRS) completion status to software. |
| 15:5 | Reserved. |
| 4 | **CrsSoftVisibilityEn: CRS software visibility enable**. Read-write. 1=Enables the root port returning configuration request retry status (CRS) completion status to software. |
| 3 | **PmIntEn: PME interrupt enable**. Read-write. 1=Enables interrupt generation upon receipt of a PME message as reflected D[8:2]F0x78[PmeStatus]. A PME interrupt is also generated if D[8:2]F0x78[PmeStatus]=1 and this bit is set by software. |

| | |
|---|---|
| 2 | **SerrOnFatalErrEn: system error on fatal error enable**. Read-write. 1=Indicates that a system error should be generated if a fatal error (ERR_FATAL) is reported by any of the devices in the hierarchy associated with this root port, or by the root port itself. |
| 1 | **SerrOnNonFatalErrEn: system error on non-fatal error enable**. Read-write. 1=Indicates that a system error should be generated if a non-fatal error (ERR_NONFATAL) is reported by any of the devices in the hierarchy associated with this root port, or by the root port itself. |
| 0 | **SerrOnCorrErrEn: system error on correctable error enable**. Read-write. 1=Indicates that a system error should be generated if a correctable error (ERR_COR) is reported by any of the devices in the hierarchy associated with this root port, or by the root port itself. |

### D[8:2]F0x78 Root Complex Status Register

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:18 | Reserved. |
| 17 | **PmePending: PME pending**. Read-only. This bit indicates that another PME is pending when PmeStatus is set. When PmeStatus is cleared by software; the PME is delivered by hardware by setting the PmeStatus bit again and updating the requestor ID appropriately. PmePending is cleared by hardware if no more PMEs are pending. |
| 16 | **PmeStatus: pme status**. Read; Write-1-to-clear. This bit indicates that PME was asserted by the requestor ID indicated in the PmeRequestorID field. Subsequent PMEs are kept pending until PmeStatus is cleared by writing a 1. |
| 15:0 | **PmeRequestorId: pme requestor ID**. Read-only. This field indicates the PCI requestor ID of the last PME requestor. |

### D[8:2]F0x7C Device Capability 2

Reset: 0000_001Fh.

| Bits | Description |
|---|---|
| 31:6 | Reserved. |
| 5 | **AriForwardingSupported**. Read-only. |
| 4 | **CplTimeoutDisSup: completion timeout disable supported**. Read-only. |
| 3:0 | **CplTimeoutRangeSup: completion timeout range supported**. Read-only. Fh=Completion timeout range is 64s to 50us. |

### D[8:2]F0x80 Device Control and Status 2

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:6 | Reserved. |
| 5 | **AriForwardingEn**. Read-only. |

| | |
|---|---|
| 4 | **CplTimeoutDis: completion timeout disable**. Read-write. 1=Completion timeout disabled. |
| 3:0 | **CplTimeoutValue: completion timeout value**. Read-write. BIOS: 6h.<br>Bits / Timeout Range: 0h 50ms-50us; 1h 100us-50us; 2h 10ms-1ms; 4h-3h Reserved; 5h 55ms-16ms; 6h 210ms-65ms; 8h-7h Reserved; 9h 900ms-260ms; Ah 3.5s-1s; Ch-Bh Reserved; Dh 13s-4s; Eh 64s-4s; Fh Reserved |

### D[8:2]F0x84 IO Link Capability 2

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:0 | Reserved. |

### D[8:2]F0x88 IO Link Control and Status 2

| Bits | Description |
|---|---|
| 31:17 | Reserved. |
| 16 | **CurDeemphasisLevel: current deemphasis level**. Read-only. Reset: D[8:2]F0xE4_xA4[LcGen2EnStrap]. 1=-3.5 dB. 0=-6 dB |
| 15:13 | Reserved. |
| 12 | **ComplianceDeemphasis: compliance deemphasis**. Read-write. Reset: 0. This bit defines the compliance deemphasis level when EnterCompliance is set. Software should leave this field in its default state. 1= -3.5 dB. 0=-6 dB |
| 11 | **ComplianceSOS: compliance SOS**. Read-write. Cold reset: 0. 1=The device transmits skip ordered sets in between the modified compliance pattern. |
| 10 | **EnterModCompliance: enter modified compliance**. Read-write. Cold reset: 0. 1=The device transmits modified compliance pattern. Software should leave this field in its default state. |
| 9:7 | **XmitMargin: transmit margin**. Read-write. Cold reset: 0. This field controls the non-deemphasized voltage level at the transmitter pins. Software should leave this field in its default state. |
| 6 | **SelectableDeemphasis: selectable deemphasis**. Read-only. Reset: D[8:2]F0xE4_xA4[LcGen2EnStrap]. 1=-3.5 dB. 0=-6 dB. |
| 5 | **HwAutonomousSpeedDisable: hardware autonomous speed disable**. Read-write. Cold reset: 0. 1=Support for hardware changing the link speed for device specific reasons disabled. |

| | |
|---|---|
| 4 | **EnterCompliance: enter compliance**. Read-write. Cold reset: 0. 1=Force the link to enter the compliance mode. |
| 3:0 | **TargetLinkSpeed: target link speed**. Read-write. Cold reset: {0b, D[8:2]F0xE4_xA4[LcGen2EnStrap],~D[8:2]F0xE4_xA4[LcGen2EnStrap]}. This fields defines the upper limit of the link operational speed. Writes of reserved encodings are not valid. Hardware prevents writes of reserved encodings from changing the state of this field.<br>Bits Definition<br>0h Reserved<br>1h 2.5GT/s<br>2h 5.0GT/s<br>Fh-3h Reserved |

### D[8:2]F0x8C Slot Capability 2

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:0 | Reserved. |

### D[8:2]F0x90 Slot Control and Status 2

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:0 | Reserved. |

### D[8:2]F0xA0 MSI Capability Register

| Bits | Description |
|---|---|
| 31:24 | Reserved. |
| 23 | **Msi64bit: MSI 64 bit capability**. Read-only. Value: D0F0x64_x46[Msi64bitEn]. 1=The device is capable of sending 64-bit MSI messages. 0=The device is not capable of sending a 64-bit message address. |
| 22:20 | **MsiMultiEn: MSI multiple message enable**. Read-write. Reset: 000b. Software writes to this field to indicate the number of allocated vectors (equal to or less than the number of requested vectors). When MSI is enabled, a function is allocated at least 1 vector. |
| 19:17 | **MsiMultiCap: MSI multiple message capability**. Read-only. Reset:000b. 000b=The device is requesting one vector. |
| 16 | **MsiEn: MSI enable**. Read-write. Reset: 0. 1=MSI generation is enabled and INTx generation is disabled. 0=MSI generation disabled and INTx generation is enabled. |
| 15:8 | **NextPtr: next pointer**. Read-only. Reset:B0h |
| 7:0 | **CapID: capability ID**. Read-only. Reset: 05h. 05h=MSI capability structure. |

### D[8:2]F0xA4 MSI Message Address

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:2 | **MsiMsgAddr: MSI message address**. Read-write. This register specifies the dword aligned address for the MSI memory write transaction. |
| 1:0 | Reserved. |

### D[8:2]F0xA8 MSI Message Data

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:16 | Reserved. |
| 15:0 | **MsiData: MSI message data**. Read-write. This register specifies lower 16 bits of data for the MSI memory write transaction. The upper 16 bits are always 0. |

### D[8:2]F0xB0 Subsystem and Subvendor Capability ID Register

Reset: 0000_B80Dh.

| Bits | Description |
|---|---|
| 31:16 | Reserved. |
| 15:8 | **NextPtr: next pointer**. Read-only. |
| 7:0 | **CapID: capability ID**. Read-only. |

### D[8:2]F0xB4 Subsystem and Subvendor ID Register

| Bits | Description |
|---|---|
| 31:16 | **SubsystemID**. Value: D0F0xE4_x013[1:0]_8002[SubsystemID]. |
| 15:0 | **SubsystemVendorID**. Value: D0F0xE4_x013[1:0]_8002[SubsystemVendorID]. |

### D[8:2]F0xB8 MSI Capability Mapping

Reset:A803_0008h.

| Bits | Description |
|---|---|
| 31:27 | **CapType: capability type**. Read-only. 19h=MSI Mapping capability. |
| 26:18 | Reserved. |
| 17 | **FixD**. Read-only. 1=Fixed MSI destination address of FEEx_xxxxh. |
| 16 | **En**. Read-only. 1=MSI mapping capability enabled. |
| 15:8 | **NextPtr: next pointer**. Read-only. |
| 7:0 | **CapID: capability ID**. Read-only. |

### D[8:2]F0xBC MSI Mapping Address Low

| Bits | Description |
|---|---|
| 31:20 | **MsiMapAddrLo**. Read-only. Reset: 0. Lower 32-bits of the MSI address. |
| 19:0 | Reserved. |

### D[8:2]F0xC0 MSI Mapping Address High

| Bits | Description |
|---|---|
| 31:0 | **MsiMapAddrHi**. Read-only. Reset: 0. Upper 32-bits of the MSI address. |

### D[8:2]F0xE0 Root Port Index

Reset: 0000_0000h.

The index/data pair registers D[8:2]F0xE0 and D[8:2]F0xE4 is used to access the registers D[8:2]F0xE4_x[FF:00]. To read or write to one of these registers, the address is written first into the address register D[8:2]F0xE0 and then the data are read or written by read or write the data register D[8:2]F0xE4.

| Bits | Description |
|---|---|
| 31:8 | Reserved. |
| 7:0 | **PcieIndex**. Read-write. |

### D[8:2]F0xE4 Root Port Data

See D[8:2]F0xE0.

### D[8:2]F0xE4_x02 Root Port Hardware Debug

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:16 | Reserved. |
| 15 | **RegsLcAllowTxL1Control**. Read-write. BIOS: 1. 1=Tx can prevent LC from entering L1 when there are outstanding completions. |
| 14:0 | Reserved. |

### D[8:2]F0xE4_x20 Root Port TX Control

Reset: 0050_8000h.

| Bits | Description |
|---|---|
| 31:16 | Reserved. |
| 15 | **TxFlushTlpDis: TLP flush disable**. Read-write. BIOS:1. 1=Disable flushing TLPs when the link is down. |
| 14:0 | Reserved. |

**D[8:2]F0xE4_x50 Root Port Lane Status**

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:7 | Reserved. |
| 6:1 | **PhyLinkWidth: port link width**. Read-only.<br>Bits / Definition / Bits / Definition<br>00_0000b disabled 00_1000b x8<br>00_0001b x1 01_0000b x12<br>00_0010b x2 10_0000b x16<br>00_0100b x4 |
| 0 | **PortLaneReversal: port lane reversal**. Read-only. 1=Port lanes order is reversed. |

**D[8:2]F0xE4_x6A Root Port Error Control**

Reset: 0000_0500h.

| Bits | Description |
|---|---|
| 31:1 | Reserved. |
| 0 | **ErrReportingDis: advanced error reporting disable**. Read-write. BIOS: 0. 1=Error reporting disabled. 0=Error reporting enabled. |

**D[8:2]F0xE4_x70 Root Port Receiver Control**

Reset: 0000_43F7h.

| Bits | Description |
|---|---|
| 31:20 | Reserved. |
| 19 | **RxRcbCplTimeoutMode: RCB completion timeout mode**. Read-write. BIOS: 1. |
| 18:16 | **RxRcbCplTimeout: RCB completion timeout**. Read-write.<br>Bits / Definition / Bits / Definition<br>000b Disabled 100b 50ms<br>001b 50us 101b 100ms<br>010b 10ms 110b 500ms<br>011b 25ms 111b 1ms |
| 15:0 | Reserved. |

**D[8:2]F0xE4_xA0 Per Port Link Controller (LC) Control**

Reset: 4000_0050h.

| Bits | Description |
|---|---|
| 31:24 | Reserved. |
| 23 | **LcL1ImmediateAck: immediate ACK ASPM L1 entry**. Read-write. BIOS: 1. 1=Always ACK ASPM L1 entry DLLPs. |
| 22:16 | Reserved. |

| | |
|---|---|
| 15:12 | **LcL1Inactivity: L1 inactivity timer**. Read-write.<br>Bits Definition Bits Definition<br>0h L1 disabled 8h 400us<br>1h 1us 9h 1ms<br>2h 2us Ah 40us<br>3h 4us Bh 10ms<br>4h 10us Ch 40ms<br>5h 20us Dh 100ms<br>6h 40us Eh 400ms<br>7h 100us Fh reserved |
| 11:8 | **LcL0sInactivity: L0s inactivity timer**. Read-write.<br>Bits Definition Bits Definition<br>0h L0s disabled 8h 4us<br>1h 40ns 9h 10us<br>2h 80ns Ah 40us<br>3h 120ns Bh 100us<br>4h 200ns Ch 400us<br>5h 400ns Dh 1ms<br>6h 1us Eh 4ms<br>7h 2us Fh reserved |
| 7:4 | **Lc16xClearTxPipe**. Read-write. BIOS: 3h. Specifies the number of clock to drain the TX pipe. |
| 3:0 | Reserved. |

### D[8:2]F0xE4_xA1 LC Training Control

Reset: 0400_1080h.

| Bits | Description |
|---|---|
| 31:13 | Reserved. |
| 12 | **LcInitSpdChgWithCsrEn: enable software initialed speed change**. Read-write. 1=Enable link speed negotiations when D[8:2]F0x68[RetrainLink]. |
| 11 | **LcDontGotoL0sifL1Armed: prevent Ls0 entry is L1 request in progress**. Read-write. BIOS: 1. 1=Prevent the LTSSM from transitioning to Rcv_L0s if an acknowledged request to enter L1 is in progress. |
| 10:0 | Reserved. |

### D[8:2]F0xE4_xA2 LC Link Width Control

Reset: 0020_0006h.

| Bits | Description |
|---|---|
| 31:23 | Reserved. |
| 22:21 | **LcDynLanesPwrState: unused link power state**. Read-write. Controls the state of unused links after a reconfiguration.<br>Bits Definition Bits Definition<br>00b on 10b SB2<br>01b SB1 11b Off |

| | |
|---|---|
| 20 | **LcUpconfigCapable: upconfigure capable**. Read-only. 1=Both ends of the link are upconfigure capable. 0=Both ends of the link are not upconfigure capable. |
| 19:14 | Reserved. |
| 13 | **LcUpconfigureDis: upconfigure disable**. Read-write. 1=Disable link upconfigure. |
| 12 | **LcUpconfigureSupport: upconfigure support**. Read-write. |
| 11 | **LcShortReconfigEn: short re-configuration enable**. Read-write. 1=Enable short link re-configuration |
| 10 | **LcRenegotiateEn: link reconfiguration enable**. Read-write. 1=Enable link re-negotiation. |
| 9 | **LcRenegotiationSupport: re-negotiation support**. Read-only. 1=Link re-negotiation is supported by the downstream device. |
| 8 | **LcReconfigNow: re-configure link**. Read-write; cleared-when-done. 1=Initiate link width change. |
| 7 | **LcReconfigArcMissingEscape**. Read-write. 1=Expedite transition from Recovery.Idle to Detect during a long reconfiguration. |
| 6:4 | **LcLinkWidthRd: current link width**. Read-only.<br>Bits / Definition / Bits / Definition<br>000b / 0 / 100b / 8<br>001b / 1 / 101b / 12<br>010b / 2 / 110b / 16<br>011b / 4 / 111b / Reserved |
| 3 | Reserved. |
| 2:0 | **LcLinkWidth: link width required**. Read-write. See: LcLinkWidthRd. |

### D[8:2]F0xE4_xA3 LC Number of FTS Control

Reset: 00FF_020Ch.

| Bits | Description |
|---|---|
| 31:10 | Reserved. |
| 9 | **LcXmitFtsBeforeRecovery: transmit FTS before recovery**. Read-write. 1=Transmit FTS before recovery. |
| 8:0 | Reserved. |

### D[8:2]F0xE4_xA4 LC Link Speed Control

Reset: 1202_8041h.

| Bits | Description |
|---|---|
| 31:30 | Reserved. |
| 29 | **LcMultUpstreamAutoSpdChngEn: enable multiple automatic speed changes**. Read-write. 1=Enable multiple automatic speed changes. |
| 28:25 | Reserved. |
| 24 | **LcOtherSideSupportsGen2: downstream link supports gen2**. Read-only. 1=The downstream link currently supports gen2. |
| 23:19 | Reserved. |

| | |
|---|---|
| 18 | **LcGoToRecovery: go to recovery**. Read-write. 1=Force link in the L0 state to transition to the recovery state. |
| 17:11 | Reserved. |
| 10 | **LcSpeedChangeAttemptFailed: speed change attempt failed**. Read-only; updated-by-hardware. 1=LcSpeedChangeAttemptsAllowed has been reached. |
| 9:8 | Reserved. |
| 7 | **LcInitiateLinkSpeedChange: initiate link speed change**. Read-write; cleared-when-done. 1=Initiate link speed negotiation. |
| 6:5 | Reserved. |
| 4 | **LcForceDisSwSpeedChange: force disable software speed changes**. Read-write. 1=Force the PCIe core to disable speed changes initiated by private registers. |
| 3:1 | Reserved. |
| 0 | **LcGen2EnStrap: gen2 PCIe support enable**. Read-write. 1=Gen2 PCIe support enabled. 0=Gen2 PCIe support disabled. |

### D[8:2]F0xE4_xA5 LC State 0

Cold reset: 0000_0000h.

**Table 72: Link controller state encodings**

| Bits | Description | Bits | Description | Bits | Description |
|---|---|---|---|---|---|
| 00h | s_Detect_Quiet. | 12h | Rcv_L0_and_Tx_L0s. | 24h | s_Rcvd_Loopback. |
| 01h | s_Start_common_Mode. | 13h | Rcv_L0_and_Tx_L0s_FTS. | 25h | s_Rcvd_Loopback_Idle. |
| 02h | s_Check_Common_Mode. | 14h | Rcv_L0s_and_Tx_L0. | 26h | s_Rcvd_Reset_Idle. |
| 03h | s_Rcvr_Detect. | 15h | Rcv_L0s_and_Tx_L0_Idle. | 27h | s_Rcvd_Disable_Entry. |
| 04h | s_No_Rcvr_Loop | 16h | Rcv_L0s_and_Tx_L0s. | 28h | s_Rcvd_Disable_Idle. |
| 05h | s_Poll_Quiet. | 17h | Rcv_L0s_and_Tx_L0s_FTS. | 29h | s_Rcvd_Disable. |
| 06h | s_Poll_Active. | 18h | s_L1_Entry. | 2Ah | s_Detect_Idle. |
| 07h | s_Poll_Compliance. | 19h | s_L1_Idle. | 2Bh | s_L23_Wait. |
| 08h | s_Poll_Config. | 1Ah | s_L1_Wait | 2Ch | Rcv_L0s_Skp_and_Tx_L0. |
| 09h | s_Config_Step1. | 1Bh | s_L1. | 2Dh | Rcv_L0s_Skp_and_Tx_L0_Idle. |
| 0Ah | s_Config_Step3. | 1Ch | s_L23_Stall. | 2Eh | Rcv_L0s_Skp_and_Tx_L0s. |
| 0Bh | s_Config_Step5. | 1Dh | s_L23_Entry. | 2Fh | Rcv_L0s_Skp_and_Tx_L0_FTS. |
| 0Ch | s_Config_Step2. | 1Eh | s_L23_Entry. | 30h | s_Config_Step2b. |
| 0Dh | s_Config_Step4. | 1Fh | s_L23_Ready. | 31h | s_Recovery_Speed. |
| 0Eh | s_Config_Step6. | 20h | s_Recovery_lock. | 32h | s_Poll_Compliance_Idle. |
| 0Fh | s_Config_Idle. | 21h | s_Recovery_Config. | 33h | s_Rcvd_Loopback_Speed. |
| 10h | Rcv_L0_and_Tx_L0. | 22h | s_Recovery_Idle. | 3Fh-34h | Reserved. |
| 11h | Rcv_L0_and_Tx_L0_Idle. | 23h | s_Training_Bit. | | |

| Bits | Description |
|---|---|
| 31:30 | Reserved. |

| | |
|---|---|
| 29:24 | **LcPrevState3: previous link state 3**. Read-only. See: Table 72. |
| 23:22 | Reserved. |
| 21:16 | **LcPrevState2: previous link state 2**. Read-only. See: Table 72. |
| 15:14 | Reserved. |
| 13:8 | **LcPrevState1: previous link state 1**. Read-only. See: Table 72. |
| 7:6 | Reserved. |
| 5:0 | **LcCurrentState: current link state**. Read-only. See: Table 72. |

### D[8:2]F0xE4_xB1 LC Control 2

Reset: 8608_0280h.

| Bits | Description |
|---|---|
| 31:21 | Reserved. |
| 20 | **LcBlockElIdleinL0: block electrical idle in l0**. Read-write. BIOS: 1. 1=Prevent electrical idle from causing the receiver to transition from L0 to L0s. |
| 19 | **LcDeassertRxEnInL0s: deassert RX_EN in L0s**. Read-write. 1=Turn off transmitters in L0s. |
| 18:0 | Reserved. |

### D[8:2]F0xE4_xB5 LC Control 3

Reset: 0050_5020h.

| Bits | Description |
|---|---|
| 31:16 | Reserved. |
| 15:14 | **LcEhpTxPhyCmd**. Read-write. |
| 13:12 | **LcEhpRxPhyCmd**. Read-write. |
| 11 | Reserved. |
| 10 | **LcEnhancedHotPlugEn: enhanced hot plug enable**. Read-write. 1=Enhanced hot plug is enabled. |
| 9:4 | Reserved. |
| 3 | **LcRcvdDeemphasis: received deemphasis**. Read-only. Deemphasis advertised by the downstream device. 1=-3.5dB. 0=-6dB. |
| 2:1 | **LcSelectDeemphasisCntl: deemphasis control**. Read-write. Specifies the deemphasis used by the transmitter.<br>Bits — Definition<br>00b — Use deemphasis from LcSelectDeemphasis.<br>01b — Use deemphasis advertised by the downstream device.<br>10b — -6dB<br>11b — -3.5dB |
| 0 | **LcSelectDeemphasis: downstream deemphasis**. Read-write. Specifies the downstream deemphasis. 1= -3.5dB. 0=-6dB. |

### D[8:2]F0xE4_xC0 LC Strap Override

| Bits | Description |
|---|---|
| 31:16 | Reserved. |
| 15 | **StrapAutoRcSpeedNegotiationDis: autonomous speed negotiation disable strap override**. Read-write. Reset: 0. BIOS: 1. |
| 14 | Reserved. |
| 13 | **StrapForceCompliance: force compliance strap override**. Read-write. Reset: 0b. |
| 12:0 | Reserved. |

### D[8:2]F0xE4_xC1 Root Port Miscellaneous Strap Override

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:5 | Reserved. |
| 4 | **StrapReverseLanes: reverse lanes strap override**. Read-write. |
| 3:0 | Reserved. |

### D[8:2]F0x100 Vendor Specific Enhanced Capability Register

| Bits | Description |
|---|---|
| 31:20 | **NextPtr: next pointer**. Read-only. IF (D0F0xE4_x0[2:1]01_00B0[StrapF0AerEn] == 1) THEN Reset: 150h.<br>ELSE Reset: 000h. ENDIF. |
| 19:16 | **CapVer: capability version**. Read-only. Reset: 1h. |
| 15:0 | **CapID: capability ID**. Read-only. Reset: 000Bh. |

### D[8:2]F0x104 Vendor Specific Header Register

Reset: 0101_0001h.

| Bits | Description |
|---|---|
| 31:20 | **VsecLen: vendor specific enhanced capability structure length**. Read-only. Defined the number of bytes of the entire vendor specific enhanced capability structure including the header. |
| 19:16 | **VsecRev: vendor specific enhanced capability version**. Read-only. |
| 15:0 | **VsecID: vendor specific enhanced capability ID**. Read-only. |

### D[8:2]F0x108 Vendor Specific 1 Register

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:0 | **Scratch: scratch**. Read-write. This field does not control any hardware. |

### D[8:2]F0x10C Vendor Specific 2 Register

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:0 | **Scratch: scratch**. Read-write. This field does not control any hardware. |

### D[8:2]F0x128 Virtual Channel 0 Resource Status Register

Reset: 0002_0000h.

| Bits | Description |
|---|---|
| 31:18 | Reserved. |
| 17 | **VcNegotiationPending: virtual channel negotiation pending**. Read-only. 1=Virtual channel negotiation in progress. This bit must be 0 before the virtual channel can be used. |
| 16 | **PortArbTableStatus: port arbitration table status**. Read-only. |
| 15:0 | Reserved. |

### D[8:2]F0x150 Advanced Error Reporting Capability

Reset: 0001_0001h.

| Bits | Description |
|---|---|
| 31:20 | **NextPtr: next pointer**. Read-only. |
| 19:16 | **CapVer: capability version**. Read-only. |
| 15:0 | **CapID: capability ID**. Read-only. |

### D[8:2]F0x154 Uncorrectable Error Status

Cold reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:26 | Reserved. |
| 25 | **TlpPrefixStatus: TLP prefix blocked status**. Read; Write-1-to-clear. |
| 24 | **AtomicOpEgressBlockedTLPStatus: atomic op egress blocked TLP status**. Read; Write-1-to-clear. |
| 23 | **McBlockedTLPStatus: MC blocked TLP status**. Read; Write-1-to-clear. |
| 22 | **UncorrInteralErrStatus: uncorrectable internal error status**. Read; Write-1-to-clear. |
| 21 | **AcsViolationStatus: access control service status**. Read; Write-1-to-clear. |
| 20 | **UnsuppReqErrStatus: unsupported request error status**. Read; Write-1-to-clear. The header of the unsupported request is logged. |
| 19 | **EcrcErrStatus: end-to-end CRC error status**. Read; Write-1-to-clear. |
| 18 | **MalTlpStatus: malformed TLP status**. Read; Write-1-to-clear. The header of the malformed TLP is logged. |
| 17 | **RcvOvflStatus: receiver overflow status**. Read-only. |

| | |
|---|---|
| 16 | **UnexpCplStatus: unexpected completion timeout status**. Read; Write-1-to-clear. The header of the unexpected completion is logged. |
| 15 | **CplAbortErrStatus: completer abort error status**. Read; Write-1-to-clear. |
| 14 | **CplTimeoutStatus: completion timeout status**. Read; Write-1-to-clear. |
| 13 | **FcErrStatus: flow control error status**. Read-only. |
| 12 | **PsnErrStatus: poisoned TLP status**. Read; Write-1-to-clear. The header of the poisoned transaction layer packet is logged. |
| 11:6 | Reserved. |
| 5 | **SurprdnErrStatus: surprise down error status**. Read-only. 0=Detection and reporting of surprise down errors is not supported. |
| 4 | **DlpErrStatus: data link protocol error status**. Read; Write-1-to-clear. |
| 3:0 | Reserved. |

### D[8:2]F0x158 Uncorrectable Error Mask

Cold reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:26 | Reserved. |
| 25 | **TlpPrefixMask: TLP prefix blocked mask**. Read-only. |
| 24 | **AtomicOpEgressBlockedTLPMask: atomic op egress blocked TLP mask**. Read-only. |
| 23 | **McBlockedTLPMask: MC blocked TLP mask**. Read-only. |
| 22 | **UncorrInteralErrMask: uncorrectable internal error mask**. Read-only. |
| 21 | **AcsViolationMask: access control service mask**. Read-only. 1=ACS violation errors are not reported. |
| 20 | **UnsuppReqErrMask: unsupported request error mask**. Read-write. 1=Unsupported request errors are not reported. |
| 19 | **EcrcErrMask: end-to-end CRC error mask**. Read-only. |
| 18 | **MalTlpMask: malformed TLP mask**. Read-write. 1=Malformed TLP errors are not reported. |
| 17 | **RcvOvflMask: receiver overflow mask**. Read-only. |
| 16 | **UnexpCplMask: unexpected completion timeout mask**. Read-write. 1=Unexpected completion errors are not reported. |
| 15 | **CplAbortErrMask: completer abort error mask**. Read-only. |
| 14 | **CplTimeoutMask: completion timeout mask**. Read-write. 1=Completion timeout errors are not reported. |
| 13 | **FcErrMask: flow control error mask**. Read-only. |
| 12 | **PsnErrMask: poisoned TLP mask**. Read-write.1=Poisoned TLP errors are not reported. |
| 11:6 | Reserved. |
| 5 | **SurprdnErrMask: surprise down error mask**. Read-only. |
| 4 | **DlpErrMask: data link protocol error mask**. Read-write. 1=Data link protocol errors are not reported. |
| 3:0 | Reserved. |

**D[8:2]F0x15C Uncorrectable Error Severity**

Cold reset: 0006_2030h.

| Bits | Description |
|---|---|
| 31:26 | Reserved. |
| 25 | **TlpPrefixSeverity: TLP prefix blocked severity**. Read-only. |
| 24 | **AtomicOpEgressBlockedTLPSeverity: atomic op egress blocked TLP severity**. Read-only. |
| 23 | **McBlockedTLPSeverity: MC blocked TLP severity**. Read-only. |
| 22 | **UncorrInteralErrSeverity: uncorrectable internal error severity**. Read-only. |
| 21 | **AcsViolationSeverity: access control service severity**. Read-only. 1=Fatal error. 0=Non-fatal error. |
| 20 | **UnsuppReqErrSeverity: unsupported request error severity**. Read-write. 1=Fatal error. 0=Non-fatal error. |
| 19 | **EcrcErrSeverity: end-to-end CRC error severity**. Read-only. |
| 18 | **MalTlpSeverity: malformed TLP severity**. Read-write. 1=Fatal error. 0=Non-fatal error. |
| 17 | **RcvOvflSeverity: receiver overflow severity**. Read-only. |
| 16 | **UnexpCplSeverity: unexpected completion timeout severity**. Read-write. I1=Fatal error. 0=Non-fatal error. |
| 15 | **CplAbortErrSeverity: completer abort error severity**. Read-only. |
| 14 | **CplTimeoutSeverity: completion timeout severity**. Read-write. I1=Fatal error. 0=Non-fatal error. |
| 13 | **FcErrSeverity: flow control error severity**. Read-only. |
| 12 | **PsnErrSeverity: poisoned TLP severity**. Read-write. 1=Fatal error. 0=Non-fatal error. |
| 11:6 | Reserved. |
| 5 | **SurprdnErrSeverity: surprise down error severity**. Read-only. |
| 4 | **DlpErrSeverity: data link protocol error severity**. Read-write. 1=Fatal error. 0=Non-fatal error. |
| 3:0 | Reserved. |

**D[8:2]F0x160 Correctable Error Status**

Cold reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:14 | Reserved. |
| 13 | **AdvisoryNonfatalErrStatus: advisory non-fatal error status**. Read; Write-1-to-clear. 1=A non-fatal unsupported request errors or a non-fatal unexpected completion errors occurred. |
| 12 | **ReplayTimerTimeoutStatus: replay timer timeout status**. Read; Write-1-to-clear. |
| 11:9 | Reserved. |
| 8 | **ReplayNumRolloverStatus: replay**. Read; Write-1-to-clear. 1=The same transaction layer packet has been replayed three times and has caused the link to re-train. |
| 7 | **BadDllpStatus: bad data link layer packet status**. Read; Write-1-to-clear. 1=A link CRC error was detected. |
| 6 | **BadTlpStatus: bad transaction layer packet status**. Read; Write-1-to-clear. 1=A bad non-duplicated sequence ID or a link CRC error was detected. |

| | |
|---|---|
| 5:1 | Reserved. |
| 0 | **RcvErrStatus: receiver error status**. Read-only. 1=An 8B10B or disparity error was detected. |

**D[8:2]F0x164 Correctable Error Mask**

Cold reset: 0000_2000h.

| Bits | Description |
|---|---|
| 31:14 | Reserved. |
| 13 | **AdvisoryNonfatalErrMask: advisory non-fatal error mask**. Read-write. 1=Error is not reported. |
| 12 | **ReplayTimerTimeoutMask: replay timer timeout mask**. Read-write. 1=Error is not reported. |
| 11:9 | Reserved. |
| 8 | **ReplayNumRolloverMask: replay**. Read-write.1=Error is not reported. |
| 7 | **BadDllpMask: bad data link layer packet mask**. Read-write. 1=Error is not reported. |
| 6 | **BadTlpMask: bad transaction layer packet mask**. Read-write. 1=Error is not reported. |
| 5:1 | Reserved. |
| 0 | **RcvErrMask: receiver error mask**. Read-only. 1=Error is not reported. |

**D[8:2]F0x168 Advanced Error Control**

Cold reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:9 | Reserved. |
| 8 | **EcrcCheckEn: data link protocol error severity**. Read-only. 0=Specifies that End-to-end CRC generation is not supported. |
| 7 | **EcrcCheckCap: data link protocol error severity**. Read-only. 0=Specifies that end-to-end CRC check is not supported. |
| 6 | **EcrcGenEn: end-to-end CRC enable**. Read-only. 0=Specifies that End-to-end CRC generation is not supported. |
| 5 | **EcrcGenCap: end-to-end CRC capability**. Read-only. 0=Specifies that end-to-end CRC generation is not supported. |
| 4:0 | **FirstErrPtr: first error pointer**. Read-only. The First Error Pointer identifies the bit position of the first error reported in the Uncorrectable Error Status register. |

**D[8:2]F0x16C Header Log DW0**

Cold reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:0 | **TlpHdr: transaction layer packet header log**. Read-only. Contains the header for a transaction layer packet corresponding to a detected error. The upper byte represents byte 0 of the header. |

### D[8:2]F0x170 Header Log DW1

Cold reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:0 | **TlpHdr: transaction layer packet header log**. Read-only. Contains the header for a transaction layer packet corresponding to a detected error. The upper byte represents byte 4 of the header. |

### D[8:2]F0x174 Header Log DW2

Cold reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:0 | **TlpHdr: transaction layer packet header log**. Read-only. Contains the header for a transaction layer packet corresponding to a detected error. The upper byte represents byte 8 of the header. |

### D[8:2]F0x178 Header Log DW3

Cold reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:0 | **TlpHdr: transaction layer packet header log**. Read-only. Contains the header for a transaction layer packet corresponding to a detected error. The upper byte represents byte 12 of the header. |

### D[8:2]F0x17C Root Error Command

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:3 | Reserved. |
| 2 | **FatalErrRepEn: fatal error reporting enable**. Read-write. 1=Enables the generation of an interrupt when a fatal error is reported by any of the devices in the hierarchy associated with this Root Port. |
| 1 | **NonfatalErrRepEn: non-fatal error reporting enable**. Read-write. 1=Enables generation of an interrupt when a non-fatal error is reported by any of the devices in the hierarchy associated with this Root Port. |
| 0 | **CorrErrRepEn: correctable error reporting enable**. Read-write. 1=Enables generation of an interrupt when a correctable error is reported by any of the devices in the hierarchy associated with this Root Port. |

### D[8:2]F0x180 Root Error Status

Cold reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:27 | **AdvErrIntMsgNum: advanced error interrupt message number**. Read-only. |
| 26:7 | Reserved. |
| 6 | **NFatalErrMsgRcvd: fatal error message received**. Read; Write-1-to-clear. Set to 1 when one or more fatal uncorrectable error messages have been received. |

| | |
|---|---|
| 5 | **NonFatalErrMsgRcvd: non-fatal error message received**. Read; Write-1-to-clear. Set to 1 when one or more non-fatal uncorrectable error messages have been received. |
| 4 | **FirstUncorrFatalRcvd: first uncorrectable fatal error message received**. Read; Write-1-to-clear. Set to 1 when the first uncorrectable error message received is for a fatal error. |
| 3 | **MultErrFatalNonfatalRcvd: ERR_FATAL/NONFATAL message received**. Read; Write-1-to-clear. Set when either a fatal or a non-fatal error is received and ErrFatalNonfatalRcvd is already set. |
| 2 | **ErrFatalNonfatalRcvd: ERR_FATAL/NONFATAL message received**. Read; Write-1-to-clear. Set when either a fatal or a non-fatal error is received and this bit is not already set. |
| 1 | **MultErrCorrRcvd: multiple ERR_COR messages received**. Read; Write-1-to-clear. Set when a correctable error message is received and ErrCorrRcvd is already set. |
| 0 | **ErrCorrRcvd: ERR_COR message received**. Read; Write-1-to-clear. Set when a correctable error message is received and this bit is not already set. |

#### D[8:2]F0x184 Error Source ID

Cold reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:16 | **ErrFatalNonfatalSrcID: ERR_FATAL/ERR_NONFATAL source identification**. Read-only. Loaded with the requestor ID indicated in the received ERR_FATAL or ERR_NONFATAL message when D[8:2]F0x180[ErrFatalNonfatalRcvd] is not already set. |
| 15:0 | **ErrCorlSrcID: ERR_COR source identification**. Read-only. Loaded with the requestor ID indicated in the received ERR_COR message when D[8:2]F0x180[ErrCorrRcvd] is not already set. |

### 3.7 Device 18h Function 0 Configuration Registers

See 3.1 [Register Descriptions and Mnemonics] for a description of the register naming convention. See 2.7 [Configuration Space] for details about how to access this space.

#### D18F0x00 Device/Vendor ID Register

Reset: 1700_1022h.

| Bits | Description |
|---|---|
| 31:16 | **DeviceID: device ID**. Read-only. |
| 15:0 | **VendorID: vendor ID**. Read-only. |

#### D18F0x04 Status/Command Register

Reset: 0010_0000h.

| Bits | Description |
|---|---|
| 31:16 | **Status**. Read-only. Bit[20] is set to indicate the existence of a PCI-defined capability block. |
| 15:0 | **Command**. Read-only. |

**D18F0x08 Class Code/Revision ID Register**

Reset: 0600_0043h.

| Bits | Description |
|---|---|
| 31:8 | **ClassCode**. Read-only. Provides the host bridge class code as defined in the PCI specification. |
| 7:0 | **RevID: revision ID**. Read-only. |

**D18F0x0C Header Type Register**

Reset: 0080_0000h.

| Bits | Description |
|---|---|
| 31:0 | **HeaderTypeReg**. Read-only. These bits are fixed at their default values. The header type field indicates that there multiple functions present in this device. |

**D18F0x34 Capabilities Pointer Register**

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:8 | Reserved. |
| 7:0 | **CapPtr: capabilities pointer**. Read-only. There is no capability block. |

**D18F0x68 Link Transaction Control Register**

| Bits | Description |
|---|---|
| 31:24 | Reserved. |
| 23 | **InstallStateS**. Read-write. Reset: 0. 1=Forces the default read block (RdBlk) install state to be shared instead of exclusive. |
| 22:21 | **DsNpReqLmt: downstream non-posted request limit**. Read-write. Reset: 0. BIOS: 01b. Specifies the maximum number of downstream non-posted requests, issued by core(s) or peer-to-peer, which may be outstanding in the root complex.<br>Bits Definition Bits Definition<br>00b no limit. 10b limited to 4.<br>01b limited to 1. 11b limited to 8. |
| 20 | **DisSeqIdReqUID: disable using requester UID for SeqID**. Read-write. Reset: 0. 0=SeqID is assigned the UID of the requesting CPU, such that CPU0=2h and CPU1=3h. 1=SeqID of 0h is used. |
| 19 | **ApicExtSpur: APIC extended spurious vector enable**. Read-write. Reset: 0. This enables the extended APIC spurious vector functionality. 0=The lower 4 bits of the spurious vector are read-only 1111b. 1=The lower 4 bits of the spurious vector are writable. |
| 18 | **ApicExtId: APIC extended ID enable**. Read-write. Reset: 0. This enables the extended APIC ID functionality. 0=APIC ID is 4 bits. 1=APIC ID is 8 bits. |
| 17 | **ApicExtBrdCst: APIC extended broadcast enable**. Read-write. Reset: 0. This enables the extended APIC broadcast functionality. 0=APIC broadcast is 0Fh. 1=APIC broadcast is FFh. |

| | |
|---|---|
| 16 | **LintEn: local interrupt conversion enable**. Read-write. Reset: 0. 1=Enables the conversion of broadcast ExtInt and NMI interrupt requests to LINT0 and LINT1 local interrupts, respectively, before delivering to the local APIC. This conversion only takes place if the local APIC is hardware enabled. 0=ExtInt/NMI interrupts delivered unchanged. |
| 15:12 | Reserved. |
| 11 | **RespPassPW: response PassPW**. Read-write. Reset: 0. BIOS: 1. 1=The PassPW bit in all downstream responses is set, regardless of the originating request packet. This technically breaks the PCI ordering rules but it is not expected to be an issue in the downstream direction. Setting this bit improves the latency of upstream requests by allowing the downstream responses to pass posted writes. 0=The PassPW bit in downstream responses is based on the RespPassPW bit of the original request. |
| 10:8 | Reserved. |
| 7 | **CPURdRspPassPW: CPU read response PassPW**. Read-write. Reset: 0. 1=Read responses to core-generated reads are allowed to pass posted writes. 0=core responses do not pass posted writes. This bit is not expected to be set. This bit may only be set during the boot process. |
| 6 | **CPUReqPassPW: CPU request PassPW**. Read-write. Reset: 0. 1=core-generated requests are allowed to pass posted writes. 0=core requests do not pass posted writes. This bit is not expected to be set. This bit may only be set during the boot process. |
| 5 | **Cpu1En: core 1 enable**. Read-write. Reset: 0. IF (D18F3xE8[CmpCap]==0) THEN BIOS: 0. ENDIF. This bit is used to enable the CPU1 core after a reset. 1=Enable core 1 to start fetching and executing code from the boot vector. |
| 4:0 | Reserved. |

**D18F0x6C Link Initialization Control Register**

| Bits | Description |
|---|---|
| 31:11 | Reserved. |
| 10:9 | **BiosRstDet[2:1]: BIOS reset detect bits[2:1]**. Read-write. Cold reset: 0. See BiosRstDet[0]. |
| 8:7 | Reserved. |
| 6 | **InitDet: CPU initialization command detect**. Read-write. Reset: 0. This bit may be used by software to distinguish between an INIT and a warm/cold reset by setting it to a 1 before an initialization event is generated. This bit is cleared by RESET_L but not by an INIT command. |
| 5 | **BiosRstDet[0]: BIOS reset detect bit[0]**. Read-write. Cold reset: 0. This bit, along with BiosRstDet[2:1], may be used to distinguish between a reset event generated by BIOS versus a reset event generated for any other reason by setting one or more of the bits to a 1 before initiating a BIOS-generated reset event. |
| 4 | **ColdRstDet: cold reset detect**. Read-write. Cold reset: 0. This bit may be used to distinguish between a cold versus a warm reset event by setting the bit to a 1 before an initialization event is generated. |
| 3:1 | Reserved. |
| 0 | Reserved. |

#### D18F0x168 Extended Link Transaction Control Register

| Bits | Description |
|---|---|
| 31:2 | Reserved. |
| 1 | **Cpu3En: CPU core 3 enable**. Read-write. See: D18F0x68[Cpu1En]. |
| 0 | **Cpu2En: CPU core 2 enable**. Read-write. See: D18F0x68[Cpu1En]. |

### 3.8 Device 18h Function 1 Configuration Registers

See 3.1 [Register Descriptions and Mnemonics] for a description of the register naming convention. See 2.7 [Configuration Space] for details about how to access this space.

#### D18F1x00 Device/Vendor ID Register

Reset: 1701_1022h.

| Bits | Description |
|---|---|
| 31:16 | **DeviceID: device ID**. Read-only. |
| 15:0 | **VendorID: vendor ID**. Read-only. |

#### D18F1x04 Status/Command Register

Read-only. Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:16 | **Status**. |
| 15:0 | **Command**. |

#### D18F1x08 Class Code/Revision ID Register

Reset: 0600_0000h.

| Bits | Description |
|---|---|
| 31:8 | **ClassCode**. Read-only. Provides the host bridge class code as defined in the PCI specification. |
| 7:0 | **RevID: revision ID**. Read-only. Processor revision. |

#### D18F1x0C Header Type Register

Reset: 0080_0000h.

| Bits | Description |
|---|---|
| 31:0 | **HeaderTypeReg**. Read-only. These bits are fixed at their default values. The header type field indicates that there multiple functions present in this device. |

### D18F1x34 Capabilities Pointer Register

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:8 | Reserved. |
| 7:0 | **CapPtr: capabilities pointer**. Read-only. |

### D18F1x40 DRAM Base Register

Reset: 0000_0000h.
These registers specify the DRAM address range, base and limit:

| Base Address | Limit Address |
|---|---|
| F1x40 | F1x44 |

DRAM mapping rules:

- Transaction addresses are within the defined range if: {DramBase[39:24], 00_0000h} <= address[39:0] <= {DramLimit[39:24], FF_FFFFh}.
- Accesses to addresses that map to both DRAM, as specified by D18F1x40, and MMIO, as specified by D18F1x[B8,B0,A8,A0,98,90,88,80], are routed to MMIO only.
- Programming of the DRAM address maps must be consistent with the Memory-Type Range Registers (MTRRs) and the top of memory registers, MSRC001_001A and MSRC001_001D. CPU accesses only hit within the DRAM address maps if the corresponding MTRR is of type DRAM. Accesses from the link are routed based on [The DRAM Base Register] D18F1x40, only.
- The appropriate RE or WE bit(s) must be set.
- See 2.8.4.1.1 [DRAM and MMIO Memory Space].

**Hoisting**. When memory hoisting is enabled (D18F1xF0[DramHoleValid]=1), D18F1x44[DramLimit] should be set up to account for the memory hoisted above the hole. I.e., D18F1x44[DramLimit] should be set to D18F1x40[DramBase] plus the size of the amount of memory plus the hole size (4G minus D18F1xF0[DramHoleBase]). See 2.9.5 [Memory Hoisting].

| Bits | Description |
|---|---|
| 31:16 | **DramBase[39:24]: DRAM base address register bits[39:24]**. Read-only. |
| 15:2 | Reserved. |
| 1 | **WE: write enable**. Read-write. 1=Writes to this address range are enabled. |
| 0 | **RE: read enable**. Read-write. 1=Reads to this address range are enabled. |

### D18F1x44 DRAM Limit Register

Reset: FFFF_0000h.See D18F1x40.

| Bits | Description |
|---|---|
| 31:16 | **DramLimit[39:24]: DRAM limit address register bits[39:24]**. IF (D18F2x118[C6DramLock] == 1) THEN Read-only. ELSE Read-write. ENDIF. |
| 15:0 | Reserved. |

### D18F1x[B8,B0,A8,A0,98,90,88,80] Memory Mapped IO Base Registers

These registers specify up to 8 MMIO address ranges. Each address range is specified by a base/limit register pair. The first set is F1x80 and F1x84, the second set is F1x88 and F1x8C, and so forth. Transaction addresses that are within the specified base/limit range are routed to the link. See [The Northbridge Routing] 2.8.4.

MMIO mapping rules:

- Transaction addresses are within the defined range if:
  {00h, MMIOBase[39:16], 0000h} <= address[39:0] <= {00h, MMIOLimit[39:16], FFFFh}.
- MMIO regions must not overlap each other.
- Accesses to addresses that map to both DRAM, as specified by D18F1x40, and MMIO, as specified by D18F1x[B8,B0,A8,A0,98,90,88,80], are routed to MMIO only.
- Programming of the MMIO address maps must be consistent with the Memory-Type Range Registers (MTRRs) and the top of memory registers, MSRC001_001A and MSRC001_001D. CPU accesses only hit within the MMIO address maps if the corresponding MTRR is of type IO. Accesses from the link routed are based on D18F1x[B8,B0,A8,A0,98,90,88,80], only.
- MMIO configuration accesses are not affected by MMIO ranges. See MSRC001_0058.
- When initializing a base/limit pair, the BIOS must write the limit register (F1x[BC,B4,AC,A4,9C,94,8C,84]) before either the RE or WE bit is set. When changing a base/limit pair that is already enabled, the BIOS should clear RE and WE before changing the address range.
- See 2.8.4.1.1 [DRAM and MMIO Memory Space].

| Bits | Description |
|---|---|
| 31:8 | **MMIOBase[39:16]: MMIO base address register bits[39:16]**. Read-write. Reset: X. |
| 7:4 | Reserved. |
| 3 | **Lock**. Read-write. Reset: X. 1=D18F1x[B8,B0,A8,A0,98,90,88,80] and D18F1x[BC,B4,AC,A4,9C,94,8C,84], are read-only (including this bit). WE or RE in this register must be set in order for this to take effect. |
| 2 | Reserved. |
| 1 | **WE: write enable**. Read-write. Reset: 0. 1=Writes to this address range are enabled. |
| 0 | **RE: read enable**. Read-write. Reset: 0. 1=Reads to this address range are enabled. |

### D18F1x[BC,B4,AC,A4,9C,94,8C,84] Memory Mapped IO Limit Registers

See D18F1x[B8,B0,A8,A0,98,90,88,80].

| Bits | Description |
|---|---|
| 31:8 | **MMIOLimit[39:16]: MMIO limit address register bits[39:16]**. Read-write. Reset: X. |

| | |
|---|---|
| 7 | **NP: non-posted**. Read-write. Reset: X. 1=CPU write requests to this MMIO range are passed through the non-posted channel. This may be used to force writes to be non-posted for MMIO regions which map to the legacy ISA/LPC bus, or in conjunction with [The Link Transaction Control Register] D18F0x68[DsNpReqLmt] in order to allow downstream CPU requests to be counted and thereby limited to a specified number. This latter use of the NP bit may be used to avoid loop deadlock scenarios in systems that implement a region in an IO device that reflects downstream accesses back upstream. See the link specification summary of deadlock scenarios for more information. 0=CPU writes to this MMIO range use the posted channel. This bit does not affect requests that come from the link (the virtual channel of the request is specified by the link request).<br><br>If two MMIO ranges target the same IO device and the NP bit is set differently in both ranges, unexpected transaction ordering effects are possible. In particular, using PCI- and IO-link-defined producer-consumer semantics, if a producer (e.g., the processor) writes data using a non-posted MMIO range followed by a flag to a posted MMIO range, then it is possible for the device to see the flag updated before the data is updated. |
| 6:0 | Reserved. |

### D18F1xC0 IO-Space Base Register

This register and D18F1xC4 specify positive decode for IO addresses to the link for transactions resulting from x86-defined IN and OUT instructions. The IO address range is specified by 1 set of base/limit registers. See [The Northbridge Routing] 2.8.4.

IO mapping rules:
- IO-space transaction addresses are within the defined range if:
  {IOBase[24:12], 000h} <= address <= {IOLimit[24:12], FFFh} and as specified by the IE bit; or
  if the address is in the range specified by the VE bits.
- The appropriate RE or WE bit(s) must be set.
- See 2.8.4.1.2 [IO Space].

| Bits | Description |
|---|---|
| 31:25 | Reserved. |
| 24:12 | **IOBase[24:12]: IO base address register bits[24:12]**. Read-write. Reset: X. |
| 11:6 | Reserved. |
| 5 | **IE: ISA enable**. Read-write. Reset: X. 1=The IO-space address window is limited to the first 256 bytes of each 1K byte block specified; this only applies to the first 64K bytes of IO space. 0=The PCI IO window is not limited in this way. |
| 4 | **VE: VGA enable**. Read-write. Reset: X. 1=Include IO-space transactions targeting the VGA-compatible address space within the IO-space window of this base/limit pair. These include IO accesses in which address bits[9:0] range from 3B0h to 3BBh or 3C0h to 3DFh (address bits[15:10] are not decoded); this only applies to the first 64K of IO space; i.e., address bits[24:16] must be low). 0=IO-space transactions targeting VGA-compatible address ranges are not added to the IO-space window. The MMIO range associated with the VGA enable bit in the PCI specification is not included in the VE bit definition; to map this range to a link, see [The VGA Enable Register] D18F1xF4. When D18F1xF4[VE] is set, the state of this bit is ignored. |
| 3:2 | Reserved. |
| 1 | **WE: write enable**. Read-write. Reset: 0. 1=Writes to this IO-space address range are enabled. |
| 0 | **RE: read enable**. Read-write. Reset: 0. 1=Reads to this IO-space address range are enabled. |

### D18F1xC4 IO-Space Limit Register

For detailed description see D18F1xC0 and 2.8.4.1.2 [IO Space].

| Bits | Description |
|---|---|
| 31:25 | Reserved. |
| 24:12 | **IOLimit[24:12]: IO limit address register bits[24:12]**. Read-write. Reset: X. |
| 11:0 | Reserved. |

### D18F1xF0 DRAM Hole Address Register

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:24 | **DramHoleBase[31:24]: DRAM hole base address**. IF (D18F2x118[C6DramLock] == 1) THEN Read-only. ELSE Read-write. ENDIF. This specifies the base address of the IO hole, below the 4G address level, that is used in memory hoisting. Normally, DramHoleBase >= MSRC001_001A[TOM[31:24]]. DramHoleBase must be > 0. See 2.9.5 [Memory Hoisting] for additional programming information. |
| 23:16 | Reserved. |
| 15:7 | **DramHoleOffset[31:23]: DRAM hole offset address**. IF (D18F2x118[C6DramLock] == 1) THEN Read-only. ELSE Read-write. ENDIF. BIOS: See 2.9.5 [Memory Hoisting].When memory hosting is enabled, this value is subtracted from the physical address of certain transactions before being passed to the DCT. See 2.9.5 [Memory Hoisting] for additional programming information. |
| 6:1 | Reserved. |
| 0 | **DramHoleValid**. IF (D18F2x118[C6DramLock] == 1) THEN Read-only. ELSE Read-write. ENDIF. 1=Memory hoisting is enabled. 0=Memory hoisting is not enabled. See 2.9.5 [Memory Hoisting] for additional programming information. |

### D18F1xF4 VGA Enable Register

Reset: 0000_0000h. Read-write. All these bits are read-write unless Lock is set.

| Bits | Description |
|---|---|

| | |
|---|---|
| 31:4 | Reserved. |
| 3 | **Lock**. Read-write. 1=All the bits in this register are read-only. |
| 2 | **CpuDis: CPU Disable**. Read-write. 1=The D18F1xF4[VE]-defined MMIO range is disabled for CPU accesses; i.e., CPU accesses to this range are treated as if the VE=0. |
| 1 | **NP: non-posted**. Read-write. 1=CPU write requests to the D18F1xF4[VE]-defined MMIO range are passed through the non-posted channel. 0=CPU writes may be posted. |
| 0 | **VE: VGA enable**. Read-write. 1=Transactions targeting the VGA-compatible address space are routed and controlled as specified by this register. 0=Transactions targeting the VGA-compatible address space are not affected by the state of this register.<br>• The VGA-compatible address space is: (1) the MMIO range A_0000h through B_FFFFh; (2) IO-space accesses in which address bits[9:0] range from 3B0h to 3BBh or 3C0h to 3DFh (address bits[15:10] are not decoded; this only applies to the first 64K of IO space; i.e., address bits[24:16] must be low).<br>• An MMIO range (F1x[BC:80]) must not overlap the VGA-compatible address space when F1xF4[VE]=1.<br>• When this bit is set, the state of D18F1xC0[VE] is ignored. |

## 3.9 Device 18h Function 2 Configuration Registers

See 3.1 [Register Descriptions and Mnemonics] for a description of the register naming convention. See 2.7 [Configuration Space] for details about how to access this space.

### D18F2x00 Device/Vendor ID Register

Reset: 1702_1022h.

| Bits | Description |
|---|---|
| 31:16 | **DeviceID: device ID**. Read-only. |
| 15:0 | **VendorID: vendor ID**. Read-only. |

### D18F2x04 Status/Command Register

Read-only. Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:16 | **Status**. |
| 15:0 | **Command**. |

### D18F2x08 Class Code/Revision ID Register

Reset: 0600_0000h.

| Bits | Description |
|---|---|
| 31:8 | **ClassCode**. Read-only. Provides the host bridge class code as defined in the PCI specification. |
| 7:0 | **RevID: revision ID**. Read-only. |

### D18F2x0C Header Type Register

Reset: 0080_0000h.

| Bits | Description |
|---|---|
| 31:0 | **HeaderTypeReg**. Read-only. These bits are fixed at their default values. The header type field indicates that there multiple functions present in this device. |

### D18F2x34 Capabilities Pointer Register

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:8 | Reserved. |
| 7:0 | **CapPtr: capabilities pointer**. Read-only. |

### D18F2x[1,0][4C:40] DRAM CS Base Address Registers

Reset: 0000_0000h.
See 2.9.1 [DCT Configuration Registers] for general programming information about DCT configuration registers.

These registers along with [The DRAM CS Mask Register] D18F2x[1,0][64:60], translate DRAM request addresses (to a DRAM controller) into DRAM chip selects. Supported DIMM sizes are specified in [The DRAM Bank Address Mapping Register] D18F2x[1,0]80. See 2.9 [DRAM Controllers (DCTs)].

For each chip select, there is a DRAM CS Base Address register. For every two chip selects there is a DRAM CS Mask Register. These are associated with DIMM numbers, CKE, and ODT signals as follows:

**Table 73: DIMM, Chip Select, CKE, ODT, and Register Mapping**

| Base Address Registers | Mask Register | DIMM Number | Chip Select | CKE | ODT |
|---|---|---|---|---|---|
| D18F2x[1, 0]40 | D18F2x[1, 0]60 | 0 | M[B,A]0_CS_L[0] | M[B,A]_CKE[0] | M[B,A]0_ODT[0] |
| D18F2x[1, 0]44 | | | M[B,A]0_CS_L[1] | M[B,A]_CKE[1] | M[B,A]0_ODT[1] |
| D18F2x[1, 0]48 | D18F2x[1, 0]64 | 1 | M[B,A]1_CS_L[0] | M[B,A]_CKE[0] | M[B,A]1_ODT[0] |
| D18F2x[1, 0]4C | | | M[B,A]1_CS_L[1] | M[B,A]_CKE[1] | M[B,A]1_ODT[1] |

The DRAM controller operates on the normalized physical address of the DRAM request. The normalized physical address includes all of the address bits that are supported by a DRAM controller. See 2.8.1 [Northbridge (NB) Architecture].

Each base address register specifies the starting normalized address of the block of memory associated with the chip select. Each mask register specifies the additional address bits that are consumed by the block of memory associated with the chip selects. If both chip selects of a DIMM are used, they must be the same size; in this case, a single mask register covers the address space consumed by both chip selects.

Lower-order address bits are provided in the base address and mask registers, as well. These allow memory to

be interleaved between chip selects, such that contiguous physical addresses map to the same DRAM page of multiple chip selects. See 2.9.4.1 [Chip Select Interleaving].

System BIOS is required to assign the largest DIMM chip select range to the lowest normalized address of the DRAM controller. As addresses increase, the chip-select size is required to remain constant or decrease. This is necessary to keep DIMM chip select banks on aligned address boundaries, regardless as to the amount of address space covered by each chip select.

For each normalized address for requests that enters a DRAM controller, a ChipSelect[i] is asserted if:

```
CSEnable[i] &
   ( {(InputAddr[36:27]  & ~AddrMask[i][36:27]),
      (InputAddr[21:13]  & ~AddrMask[i][21:13])} ==
     {(BaseAddr[36:27]    & ~AddrMask[i][36:27]),
      (BaseAddr[21:13]    & ~AddrMask[i][21:13])} );
```

| Bits | Description |
|---|---|
| 31:29 | Reserved. |
| 28:19 | **BaseAddr[36:27]: normalized physical base address bits [36:27]**. IF (D18F2x118[C6DramLock] == 1) THEN Read-only. ELSE Read-write. ENDIF. |
| 18:14 | Reserved. |
| 13:5 | **BaseAddr[21:13]: normalized physical base address bits [21:13]**. IF (D18F2x118[C6DramLock] == 1) THEN Read-only. ELSE Read-write. ENDIF. |
| 4 | Reserved. |
| 3 | **OnDimmMirror: on-DIMM mirroring (ODM) enabled**. IF (D18F2x118[C6DramLock] == 1) THEN Read-only. ELSE Read-write. ENDIF. 1=Address and bank bits are swapped by hardware for MRS commands sent to this chip select. This mode accounts for routing on the DIMM. This bit is expected to be set for the odd numbered rank of unbuffered DDR3 DIMMs if SPD byte 63 indicates that address mapping is mirrored. Hardware bit swapping does not occur for commands sent via D18F2x[1,0]7C[SendMrsCmd] when D18F2x[1,0]7C[EnDramInit] = 0. See 2.9.3.6.1.1 [DDR3 MR Initialization].<br>The following bits are swapped when enabled:<br>• M[B, A]_BANK[0] and M[B, A]_BANK[1].<br>• M[B, A]_ADD[3] and M[B, A]_ADD[4].<br>• M[B, A]_ADD[5] and M[B, A]_ADD[6].<br>• M[B, A]_ADD[7] and M[B, A]_ADD[8].<br>This bit must be programmed properly before initializing the DRAM devices. |
| 2 | **TestFail: memory test failed**. IF (D18F2x118[C6DramLock] == 1) THEN Read-only. ELSE Read-write. ENDIF. Set by BIOS to indicate that a rank is present but has failed memory training or a memory consistency test, indicating that the memory is bad. BIOS should treat CSEnable=1 and TestFail=1 as mutually exclusive. |
| 1 | Reserved. |
| 0 | **CSEnable: chip select enable**. IF (D18F2x118[C6DramLock] == 1) THEN Read-only. ELSE Read-write. ENDIF. Software may only change the value of this field before performing the sequence described in 2.9.3.6 [DRAM Device Initialization], or when DRAM is in self-refresh. See D18F2x[1,0]90[EnterSelfRef]. |

### D18F2x[1,0][64:60] DRAM CS Mask Register

Reset: 0000_0000h.
See 2.9.1 [DCT Configuration Registers] for general programming information about DCT configuration registers. See D18F2x[1,0][4C:40] for information about this register.

| Bits | Description |
|---|---|
| 31:29 | Reserved. |
| 28:19 | **AddrMask[36:27]: normalized physical address mask bits [36:27]**. IF (D18F2x118[C6DramLock] == 1) THEN Read-only. ELSE Read-write. ENDIF. |
| 18:14 | Reserved. |
| 13:5 | **AddrMask[21:13]: normalized physical address mask bits [21:13]**. IF (D18F2x118[C6DramLock] == 1) THEN Read-only. ELSE Read-write. ENDIF. |
| 4:0 | Reserved. |

### D18F2x[1,0]78 DRAM Control Register

See 2.9.1 [DCT Configuration Registers] for general programming information about DCT configuration registers.

| Bits | Description |
|---|---|
| 31:22 | **MaxRdLatency: maximum read latency**. Read-write. IF (REG==D18F2x178) THEN Reset: 0 Reset: 12h.BIOS: See 2.9.3.7.5. The DRAM controller uses this field to determine when incoming DRAM read data can be safely transferred to the DRAM controller clock (NCLK) domain. The time includes the asynchronous and synchronous latencies.<br>Bits — Definition<br>004h-000h — Reserved<br>050h-005h — <MaxRdLatency> NCLKs<br>3FFh-051h — Reserved |
| 21 | **DisCutThroughMode: disable cut through mode**. Read-write. Reset: 1. BIOS: 0. 1=DRAM controller waits for all 512 bits of DRAM read data before transferring data to DRAM controller clock (NCLK) domain. 0=DRAM controller waits for first 256 bits of DRAM read data before transferring. See 2.9.3.7.5 [Calculating MaxRdLatency]. |
| 20 | **ForceCasToSlot: force CAS to slot**. Read-write. Reset: 0. BIOS: See 2.9.3.4.7 and 2.9.3.7.5.1. 1=DRAM controller will issue all CAS commands in a specified slot of an unskipped DRAM controller clock (NCLK). 0=DRAM controller can issue CAS commands in either slot0 or slot1. The slot is specified by SlotSel. |
| 19 | **SlotSel: slot select**. Read-write. Reset: 0. BIOS: See 2.9.3.4.7 and 2.9.3.7.5.1. This field is only valid when ForceCasToSlot=1. 1=DRAM controller will issue all CAS commands in slot1 of an unskipped DRAM controller clock (NCLK). 0=DRAM controller will issue all CAS commands in slot0. |
| 18 | Reserved. |
| 17 | **AddrCmdTriEn: address command tristate enable**. Read-write. Reset: 0. BIOS: See 2.9.3.5. 1=Tristate the address, command, and bank pins when a Deselect command is issued. |
| 16 | Reserved. |

| | |
|---|---|
| 15 | **MaxSkipErrTrain: maximum skip error training**. Read-write. Reset: 0. BIOS: See 2.9.3.4.7 and 2.9.3.7.5.1. This field should only be 1 during DRAM training. 1=DRAM controller will schedule CAS commands only when separation between transmit FIFO read and write pointers are maximum value. 0=DRAM controller will schedule CAS commands as normal. If D18F2x[1,0]88[Tcl]<= 2, MaxSkipErrTrain must be 0. |
| 14 | **Slot1ExtraClkEn: slot 1 extra clock enable**. Read-write. Reset: 1. BIOS: See 2.9.3.4.7 and 2.9.3.7.5.1. 1=DRAM controller will use MaxRdLatency+1 for CAS commands scheduled in slot 1 of an unskipped DRAM controller clock (NCLK). 0=DRAM controller will use MaxRdLatency for CAS commands in slot1. |
| 13:12 | **Trdrd[3:2]: read to read timing**. Read-write. Reset: 0. This field along with D18F2x[1,0]8C[Trdrd[1:0]] combine to specify a 4-bit value, Trdrd[3:0]. See: D18F2x[1,0]8C[Trdrd[1:0]]. |
| 11:10 | **Twrwr[3:2]: write to write timing**. Read-write. Reset: 0. This field along with D18F2x[1,0]8C[Twrwr[1:0]] combine to specify a 4-bit value, Twrwr[3:0]. See: D18F2x[1,0]8C[Twrwr[1:0]]. |
| 9:8 | **Twrrd[3:2]: write to read DIMM termination turnaround**. Read-write. Reset: 0. This field along with D18F2x[1,0]8C[Twrrd[1:0]] combine to specify a 4-bit value, Twrrd[3:0]. See: D18F2x[1,0]8C[Twrrd[1:0]]. |
| 7 | Reserved. |
| 6 | **RxPtrInitReq: receive FIFO pointer initialization request**. Read; write-1-only; cleared-when-done. Reset: 0. 1=The DCT performs receive FIFO pointer initialization. This bit is cleared by hardware after the initialization completes. |
| 5:4 | Reserved. |
| 3:0 | **RdPtrInit: read pointer initial value**. Read-write. Reset: 6h. BIOS: See 2.9.3.2.2. There is a synchronization FIFO between the NB clock domain and the memory clock domain. This field specifies where the read pointer is placed at the time of FIFO initialization. This field along with D18F2x[1,0]94[MemClkFreq] gives an offset from the minimum read to write pointer separation as follows:<br>Bits — Definition<br>0010b-0000b — Reserved<br>0011b — IF (D18F2x[1,0]A8[DbeGskMemClkAlignMode]==01) THEN Reserved ELSE Minimum separation + 2.0 MEMCLKs ENDIF.<br>0100b — Minimum separation + 1.5 MEMCLKs<br>0101b — Minimum separation + 1 MEMCLK<br>0110b — Minimum separation + 0.5 MEMCLK<br>0111b — Minimum separation<br>1111b-1000b — Reserved |

### D18F2x[1,0]7C DRAM Initialization Register

Reset: 0000_0000h.

BIOS can directly control the DRAM initialization sequence using this register. To do so, BIOS sets EnDramInit to start DRAM initialization. BIOS should then complete the initialization sequence specified in the appropriate JEDEC specification. After completing the sequence, BIOS clears EnDramInit to complete DRAM

initialization.

Setting more than one of the command bits in this register (SendZQCmd, SendMrsCmd, SendAutoRefresh, and SendPchgAll) at a time results in undefined behavior. See 2.9.3 [DCT/DRAM Initialization and Resume].

| Bits | Description |
|---|---|
| 31 | **EnDramInit: enable DRAM initialization**. Read-write. BIOS: See 2.9.3.2 and 2.9.3.6.1. 1=Place the DRAM controller in BIOS-controlled DRAM initialization mode. The DCT asserts memory reset and deasserts CKE when this bit is set. |
| 30 | Reserved. |
| 29 | **SendZQCmd: send ZQ command**. Read; write-1-only; cleared-when-done. 1=The DCT sends the ZQ calibration command to all enabled chip selects. This bit is cleared by the hardware after the command completes. |
| 28 | **AssertCke: assert CKE**. Read-write. Setting this bit causes the DCT to assert the CKE pins. This bit cannot be used to deassert the CKE pins. |
| 27 | **DeassertMemRstX: deassert memory reset**. Read-write. Setting this bit causes the DCT to deassert the memory reset. This bit cannot be used to assert the memory reset pin. |
| 26 | **SendMrsCmd: send MRS/EMRS command**. Read; Write-1-only; cleared-when-done. 1=The DCT sends the MRS or EMRS commands defined by the MrsAddress and MrsBank fields of this register. Software is responsible for DRAM timing parameter enforcement. |
| 25 | **SendAutoRefresh: send auto refresh command**. Read; Write-1-only; cleared-when-done. 1=The DCT sends an auto refresh command. Only valid when EnDramInit=1. |
| 24 | **SendPchgAll: send precharge all command**. Read; Write-1-only; cleared-when-done. 1=The DCT sends a precharge-all command. Only valid when EnDramInit=1. |
| 23 | Reserved. |
| 22:20 | **MrsChipSel: MRS/EMRS command chip select**. Read-write. This field specifies which DRAM chip select is used for MRS/EMRS commands. This field is valid only when EnDramInit = 0 and when SendMrsCmd=1; otherwise, MRS/EMRS commands are sent to all chip selects.<br>Bits — Definition<br>000b — MRS/EMRS command is sent to CS0<br>001b — MRS/EMRS command is sent to CS1<br>010b — MRS/EMRS command is sent to CS2<br>011b — MRS/EMRS command is sent to CS3<br>111b-100b — Reserved. |
| 19 | Reserved. |
| 18:16 | **MrsBank: bank address for MRS/EMRS commands**. Read-write. This field specifies the data driven on the DRAM bank pins for MRS and EMRS commands. |
| 15:0 | **MrsAddress: address for MRS/EMRS commands**. Read-write. This field specifies the data driven on the DRAM address pins 15-0 for MRS and EMRS commands. |

### D18F2x[1,0]80 DRAM Bank Address Mapping Register

Reset: 0000_0000h.See 2.9.1 [DCT Configuration Registers] for general programming information about DCT configuration registers. These fields specify DIMM configuration information. These fields are required to be programmed per the following table, based on the DRAM device size and width information of the DIMM. Table 74 shows the bit numbers for each position.

| Bits | Description |
|---|---|
| 31:8 | Reserved. |
| 7:4 | **Dimm1AddrMap: DIMM 1 address map**. IF (D18F2x118[C6DramLock] == 1) THEN Read-only. ELSE Read-write. ENDIF. |
| 3:0 | **Dimm0AddrMap: DIMM 0 address map**. IF (D18F2x118[C6DramLock] == 1) THEN Read-only. ELSE Read-write. ENDIF. |

**Table 74: DDR3 DRAM address mapping**

| Bits | CS Size | Device size, width | Bank | | | Address | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 2 | 1 | 0 | | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0000b | | Reserved | | | | Row | | | | | | | | | | | | | | | | |
| | | | | | | Col | | | | | | | | | | | | | | | | |
| 0001b | 256MB | 512Mb, x16 | 15 | 14 | 13 | Row | x | x | x | x | 17 | 16 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 |
| | | | | | | Col | x | x | x | x | x | AP | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 |
| 0010b | 512MB | 512Mb, x8 | 15 | 14 | 13 | Row | x | x | x | 17 | 16 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 |
| | | 1Gb, x16 | | | | Col | x | x | x | x | x | AP | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 |
| 0011b | | Reserved | | | | Row | | | | | | | | | | | | | | | | |
| | | | | | | Col | | | | | | | | | | | | | | | | |
| 0100b | | Reserved | | | | Row | | | | | | | | | | | | | | | | |
| | | | | | | Col | | | | | | | | | | | | | | | | |
| 0101b | 1GB | 1Gb, x8 | 15 | 14 | 13 | Row | x | x | 17 | 16 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 |
| | | 2Gb, x16 | | | | Col | x | x | x | x | x | AP | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 |
| 0110b | | Reserved | | | | Row | x | | | | | | | | | | | | | | | |
| | | | | | | Col | x | | | | | | | | | | | | | | | |
| 0111b | 2GB | 2Gb, x8 | 15 | 14 | 13 | Row | x | 17 | 16 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 |
| | | 4Gb, x16 | | | | Col | x | x | x | x | x | AP | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 |
| 1000b | | Reserved | | | | Row | | | | | | | | | | | | | | | | |
| | | | | | | Col | | | | | | | | | | | | | | | | |
| 1001b | | Reserved | | | | Row | | | | | | | | | | | | | | | | |
| | | | | | | Col | | | | | | | | | | | | | | | | |
| 1010b | 4GB | 4Gb, x8 | 15 | 14 | 13 | Row | 17 | 16 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 |
| | | 8Gb, x16 | | | | Col | x | x | x | x | x | AP | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 |
| 1011b | 8GB | 8Gb, x8 | 16 | 15 | 14 | Row | 17 | 32 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 |
| | | | | | | Col | x | x | x | x | 13 | AP | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 |

### D18F2x[1,0]84 DRAM MRS Register

Reset: 0000_0001h. All the fields of this register are programmed into the DRAM device mode registers, MR[3, 2, 1, 0], for each DRAM device during the DRAM initialization process.

| Bits | Description |
|---|---|
| 31:24 | Reserved. |

| | |
|---|---|
| 23 | **PchgPDModeSel: precharge power down mode select**. Read-write. BIOS: 1. 0=DDR3-defined slow exit mode; the DCT issues the first valid read, read with auto-precharge, or synchronous ODT command a minimum of Txpdll after precharge power down exit. 1=DDR3-defined fast exit mode; the DCT issues the first valid command a minimum of Txp after precharge power down exit. |
| 22:20 | **Tcwl: CAS write latency**. Read-write. BIOS: See 2.9.3.3. This specifies the number of clock cycles from internal write command to first write data in.<br>Bits Definition Bits Definition<br>100b-000b <Tcwl+5> clocks 111b-101b Reserved |
| 19:7 | Reserved. |
| 6:4 | **Twr: write recovery**. Read-write. This specifies the minimum time from the last data write until the chip select bank precharge; this is the WR field in the DDR3 specification.<br>Bits Definition Bits Definition<br>000b 16 MEMCLK cycles 100b 8 MEMCLK cycles<br>001b 5 MEMCLK cycles 101b 10 MEMCLK cycles<br>010b 6 MEMCLK cycles 110b 12 MEMCLK cycles<br>011b 7 MEMCLK cycles 111b 14 MEMCLK cycles |
| 3:2 | Reserved. |
| 1:0 | **BurstCtrl: burst length control**. Read-write. Specifies the number of sequential beats of DQ related to one read or write command.<br>Bits Definition<br>00b Reserved.<br>01b 4 or 8-beat burst length on the fly; one 32-byte access or one 64-byte access<br>1xb Reserved |

### D18F2x[1,0]88 DRAM Timing Low Register

Reset: FF00_0001h. See 2.9.1 [DCT Configuration Registers] for general programming information about DCT configuration registers.

| Bits | Description |
|---|---|
| 31:24 | **MemClkDis: MEMCLK disable**. Read-write. BIOS: See 2.9.3.9. 1=Disable the MEMCLK. The bits MemClkDis[7:0] are mapped to package pin names as follows:<br>Bit Package pin name<br>[0] M[B,A]_CLK_H/L[0].<br>[1] M[B,A]_CLK_H/L[1].<br>[2] M[B,A]_CLK_H/L[2] (FM1 package only).<br>[3] M[B,A]_CLK_H/L[3] (FM1 package only).<br>[7:4] Reserved. |
| 23:4 | Reserved. |
| 3:0 | **Tcl: CAS latency**. Read-write. BIOS: See 2.9.3.3. This specifies the time from the CAS assertion for a read cycle until data return (from the perspective of the DRAM devices).See D18F2x[1,0]78[MaxSkipErrTrain].<br>Bits Definition<br>0h Reserved<br>Ah-1h <Tcl+4> clocks<br>Fh-Bh Reserved |

### D18F2x[1,0]8C DRAM Timing High Register

Reset: 0000_000Ah.

See 2.8.2 [DCT Configuration Registers] on page 14 for general programming information about DCT configuration registers.

<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>31:26</td><td>Reserved.</td></tr>
<tr><td>25:23</td><td>Trfc1: auto-refresh row cycle time for DIMM1. Read-write. See: Trfc0.</td></tr>
<tr><td>22:20</td><td>Trfc0: auto-refresh row cycle time for DIMM0. Read-write. BIOS: See 2.9.3.3. This specifies the minimum time from an auto-refresh command to the next valid command, except NOP or DES. DIMM numbers are specified by [The DRAM CS Base Address Registers] D18F2x[1,0][4C:40]. The recommended programming of this register varies based on DRAM density and speed.
<table>
<tr><th>Bits</th><th>Definition</th><th>Bits</th><th>Definition</th></tr>
<tr><td>000b</td><td>Reserved</td><td>100b</td><td>300 ns (all speeds, 4 Gbit)</td></tr>
<tr><td>001b</td><td>90 ns (all speeds, 512 Mbit)</td><td>101b</td><td>350 ns (all speeds, 8 Gbit)</td></tr>
<tr><td>010b</td><td>110 ns (all speeds, 1 Gbit)</td><td>11xb</td><td>Reserved</td></tr>
<tr><td>011b</td><td>160 ns (all speeds, 2 Gbit)</td><td></td><td></td></tr>
</table></td></tr>
<tr><td>19</td><td>Reserved.</td></tr>
<tr><td>18</td><td>DisAutoRefresh: disable automatic refresh. Read-write. BIOS: See 2.9.3.51=Automatic refresh is disabled. BIOS must set this bit prior to DRAM initialization and it must remain set until DRAM training has completed. Subsequent register accesses may only set this bit during S3 exit or if the DRAM has been placed into self-refresh.</td></tr>
<tr><td>17:16</td><td>Tref: refresh rate. Read-write. BIOS: See 2.9.3.3. This specifies the average time between refresh requests to all DRAM devices.
<table>
<tr><th>Bits</th><th>Definition</th></tr>
<tr><td>00b</td><td>Undefined behavior.</td></tr>
<tr><td>01b</td><td>Reserved</td></tr>
<tr><td>10b</td><td>Every 7.8 microseconds.</td></tr>
<tr><td>11b</td><td>Every 3.9 microseconds.</td></tr>
</table></td></tr>
<tr><td>15:14</td><td>Trdrd[1:0]: read-to-read timing. Read-write. BIOS: See 2.9.3.4.1 [Trdrd and TrdrdSD (Read-to-Read Timing)].Trdrd specifies the minimum number of cycles from the last clock of virtual CAS of a first read-burst operation to the clock in which CAS is asserted for a following read-burst operation to a different DIMM. This field along with D18F2x[1,0]78[Trdrd[3:2]] combine to specify a 4-bit value, Trdrd[3:0].
<table>
<tr><th>Bits</th><th>Definition</th></tr>
<tr><td>8h-0h</td><td><Trdrd+2> clocks</td></tr>
<tr><td>Fh-9h</td><td>Reserved</td></tr>
</table></td></tr>
<tr><td>13:12</td><td>Twrwr[1:0]: write-to-write timing. Read-write. BIOS: See 2.9.3.4.2 [Twrwr and TwrwrSD (Write-to-Write Timing)]. Twrwr specifies the minimum number of cycles from the last clock of virtual CAS of the first write-burst operation to the clock in which CAS is asserted for a following write-burst operation to a different DIMM. This field along with D18F2x[1,0]78[Twrwr[3:2]] combine to specify a 4-bit value, Twrwr[3:0].
<table>
<tr><th>Bits</th><th>Definition</th></tr>
<tr><td>9h-0h</td><td><Twrwr+1> clocks</td></tr>
<tr><td>Fh-Ah</td><td>Reserved</td></tr>
</table></td></tr>
</table>

| | |
|---|---|
| 11:10 | **Twrrd[1:0]: write-to-read DIMM termination turnaround**. Read-write. BIOS: See 2.9.3.4.3 [Twrrd and TwrrdSD (Write-to-Read DIMM Termination Turn-around)]. This specifies the minimum number of cycles from the last clock of *virtual CAS* of the first write operation to the clock in which CAS is asserted for a following read operation to a different DIMM. This field along with D18F2x[1,0]78[Twrrd[3:2]] combine to specify a 4-bit value, Twrrd[3:0].<br>Bits Definition<br>Ah-0h <Twrrd+1> clocks<br>Fh-Bh Reserved |
| 9:8 | Reserved. |
| 7:4 | **TrwtTO: read-to-write turnaround for data, DQS contention**. Read-write. BIOS: See 2.9.3.4.4 [TrwtTO (Read-to-Write Turnaround for Data, DQS Contention)]. This specifies the minimum number of cycles from the last clock of *virtual CAS* of a first read operation to the clock in which CAS is asserted for a following write operation. Time may need to be inserted to ensure there is no bus contention on bidirectional pins.<br>Bits Definition<br>0h Reserved<br>Fh-1h <TrwtTO+2> clocks (<TrwtTO> idle cycles on data bus)<br>See [The TrwtTO (Read-to-Write Turnaround for Data, DQS Contention)] 2.9.3.4.4. |
| 3:0 | **TrwtWB: read-to-write turnaround for opportunistic write bursting**. Read-write. BIOS: 4h. Specifies the minimum number of NCLK cycles from the last read operation seen by the DCT scheduler to the following write operation. The purpose of this field is to hold off write operations until several cycles have elapsed without a read cycle; this may result in a performance benefit. If opportunistic write bursting is disabled then DCT write bursting should be disabled by setting D18F2x11C[DctWrLimit] to 1Fh.<br>Bits Definition<br>0h disabled<br>Fh-1h <TrwtWB> NCLKs |

### D18F2x[1,0]90 DRAM Configuration Low Register

See 2.9.1 [DCT Configuration Registers] for general programming information about DCT configuration registers.

| Bits | Description |
|---|---|
| 31:30 | Reserved. |
| 29 | Reserved. |
| 28 | Reserved. |
| 27 | **DisDllShutdownSR: disable DLL shutdown in self-refresh mode**. Read-write. Reset: 0. BIOS: 0. See 2.5.6.1 [DRAM Self-Refresh]. 1=DDR phy DLLs remain active during DRAM self refresh. 0=DDR phy DLLs are shutdown during self-refresh. This bit must be programmed the same on both DCTs. |
| 26 | **DbeSkidBufDis: disable skid buffer**. Read-write. Reset: 0. BIOS: See 2.9.3.5. 1=Disable the performance enhancing skid buffer and arbitrate normally. The skid buffer allows the arbiter to pick a lower relative priority page miss ahead of a page hit, so that Trcd penalty for subsequent CAS is hidden behind the ready/requested CAS. |
| 25 | **EnDispAutoPrecharge: enable auto-precharge for display traffic**. Read-write. Reset: 0. BIOS: 1. 1=Auto-precharge commands are generated for the last read or write of a display burst if no other commands are pending to the DRAM page. |

| | |
|---|---|
| 24 | Reserved. |
| 23 | **ForceAutoPchg: force auto precharging**. Read-write. Reset: 0. BIOS: See 2.9.3.5. 1=Force auto-precharge cycles with every read or write command. This may be preferred in situations where power savings is favored over performance. |
| 22:21 | **IdleCycInit: idle cycle counter initial value**. Read-write. Reset: 0. BIOS: 11b. This specifies the initial number of MEMCLK cycles during which an open page of DRAM is not accessed before it may be closed by the dynamic page close logic. This field is ignored if D18F2x[1,0]90[DynPageCloseEn] = 0.<br>Bits / Definition / Bits / Definition<br>00b / 16 clocks / 10b / 64 clocks<br>01b / 32 clocks / 11b / 96 clocks |
| 20 | **DynPageCloseEn: dynamic page close enable**. Read-write. Reset: 0. BIOS: See 2.9.3.5.1=The DRAM controller dynamically determines when to close open pages based on the history of that particular page and D18F2x[1,0]90[IdleCycInit]. 0=Any open pages not auto-precharged by the DRAM controller are automatically closed after a time controlled by D18F6x[1,0]74. |
| 19:18 | Reserved. |
| 17 | **EnterSelfRef: enter self refresh command**. Read; write-1-only. Reset: 0. 1=Command the DRAMs to enter into self refresh mode. 0=The enter-self-refresh command has completed executing. See DisDllShutdownSR and 2.9.3.7 [DRAM Training]. NB P-state transitions must be disabled prior to setting this bit. See D18F6x90[NbPsCtrlDis] and See 2.5.4.1.3 [Software Controlled NB P-states]. |
| 16:2 | Reserved. |
| 1 | **ExitSelfRef: exit self refresh command**. Read; write-1-only. Reset: 0. 1=Command the DRAM controller to bring the DRAMs out of self refresh mode. 0=The exit-self-refresh command has completed. This command should be executed by BIOS when returning from the suspend to RAM state after the DRAM controller configuration registers are properly initialized (see 2.5.7.1.1 [ACPI Suspend to RAM State (S3)]), or when self refresh is used during DRAM training (see DisDllShutdownSR and 2.9.3.7 [DRAM Training]). This bit should not be set if the DCT is disabled. |
| 0 | Reserved. |

### D18F2x[1,0]94 DRAM Configuration High Register

Reset: 0F00_0000h.

See 2.9.1 [DCT Configuration Registers] for general programming information about DCT configuration registers.

| Bits | Description |
|---|---|
| 31:28 | **FourActWindow: four bank activate window**. Read-write. BIOS: See 2.9.3.3.1. Specifies the rolling tFAW window during which no more than 4 banks in an 8-bank device are activated.<br>Bits / Window size<br>0h / No tFAW window restriction.<br>Dh-1h / <2*FourActWindow+14> MEMCLK cycles<br>Fh-Eh / Reserved |

| | |
|---|---|
| 27:24 | **DcqBypassMax: DRAM controller queue bypass maximum**. Read-write. BIOS: Eh. The DRAM controller arbiter normally allows transactions to pass other transactions in order to optimize DRAM bandwidth. This field specifies the maximum number of times that the oldest memory-access request in the DRAM controller queue may be bypassed before the arbiter decision is overridden and the oldest memory-access request is serviced instead.<br>Bits Definition<br>0h Reserved.<br>Eh-1h The oldest request may be bypassed no more than <4*DcqBypassMax+4> times.<br>Fh The bypass maximum control is disabled. |
| 23 | **ProcOdtDis: processor on-die termination disable**. Read-write. 1=The processor-side on-die termination is disabled. 0=Processor-side on-die termination enabled. See D18F2x[1,0]9C_x0000_0000[ProcOdt] for ODT definitions. |
| 22 | **BankSwizzleMode: bank swizzle mode**. Read-write. BIOS: See 2.9.3.5. 1=Remaps the DRAM device bank address bits as a function of normalized physical address bits. Each of the bank address bits, as specified in Table 74 of D18F2x[1,0]80, are remapped as follows:<br><br>Define X as a bank address bit (e.g., X=15 if the bank bit is specified to be address bit 15).<br>Define S(n) as the state of address bit n (0 or 1) and B as the remapped bank address bit. Then,<br>B= S(X) ^ S(X + 3) ^ S(X + 6); for an 8-bank DRAM.<br><br>For example, encoding 02h of Table 74 would be remapped from bank[2:0]={A15, A14, A13} to the following for a 64-bit DCT: Bank[2:0] = {A15 ^ A18 ^ A21, A14 ^ A17 ^ A20, A13 ^ A16 ^ A19}. |
| 21 | Reserved. |
| 20 | **SlowAccessMode: slow access mode (a.k.a. 2T mode)**. Read-write. BIOS: Table 23 through Table 24. 1=One additional MEMCLK of setup time is provided on all DRAM address and control signals (not including CS, CKE, and ODT); i.e., these signals are driven for two MEMCLK cycles rather than one. 0=DRAM address and control signals are driven for one MEMCLK cycle. 2T mode may be needed in order to meet electrical requirements of certain DIMM speed and loading configurations. |
| 19:17 | Reserved. |
| 16 | **PowerDownMode: power down mode**. Read-write. BIOS: 1.This specifies how a DIMM enters power down mode, when enabled by D18F2x[1,0]94[PowerDownEn]. A DIMM enters power down mode when the DCT deasserts the CKE pin to that DIMM. The command and address signals tristate one MEMCLK after CKE deasserts. There are two CKE pins per DRAM channel. For each channel:<br>Bit Description<br>0b Channel CKE control mode. The DRAM channel is placed in power down mode when all chip selects associated with the channel are idle. Both CKE pins for the channel operate in lock step, in terms of placing the channel DIMMs in power down mode.<br>1b Chip select CKE control mode. A chip select or pair of chip selects is placed in power down mode when no transactions are pending for the chip select(s). This mode is expected to be used in mobile systems:<br>- CKE0 is associated with CS0 in 2-rank systems.<br>- CKE1 is associated with CS1 in 2-rank systems.<br>• See D18F2x[1,0][4C:40]. |
| 15 | **PowerDownEn: power down mode enable**. Read-write. BIOS: See 2.9.3.5. 1=Power down mode is enabled. When in power down mode, if all pages of the DRAMs associated with a CKE pin are closed, then these parts are placed in power down mode. Only pre-charge power down mode is supported, not active power down mode. |

| | |
|---|---|
| 14 | **DisDramInterface: disable the DRAM interface**. Read-write. BIOS: See 2.9.3.8. 1=The DRAM controller is disabled and the DRAM interface is placed into a low power state. This bit must be set if there are no DIMMs connected to the DCT. If this bit is set, BIOS must also disable the phy for the channel. See D18F2x[1,0]9C_x0000_000B[DynModeChange]. If this bit is set, BIOS must program D18F2x[1,0]9C_x0000_000C[CKETri]=11b for the channel a minimum of 24 MEMCLKs before setting DisDramInterface or D18F2x[1,0]9C_x0000_000B[DynModeChange]. |
| 13:12 | Reserved. |
| 11:10 | **ZqcsInterval: ZQ calibration short interval**. Read-write. BIOS: See 2.9.3.5. This field specifies the programmable interval for the controller to send out the DRAM ZQ calibration short command.<br>Bits — Definition<br>00b — ZQ calibration short command is disabled<br>01b — 64 ms<br>10b — 128 ms<br>11b — 256 ms |
| 9:8 | Reserved. |
| 7 | **MemClkFreqVal: memory clock frequency valid**. Read-write. System BIOS should set this bit when setting up D18F2x[1,0]94[MemClkFreq] to the proper value. This indicates to the DRAM controller that it may start driving MEMCLK at the proper frequency. This bit should not be set if the DCT is disabled. BIOS must change each DCT's operating frequency in order. See 2.9.3.6. |
| 6:5 | Reserved. |
| 4:0 | **MemClkFreq: memory clock frequency**. Read-write. This field specifies the frequency and rate of the DRAM interface (MEMCLK). The rate defined below is twice the frequency. See D18F3xE8[DdrMaxRate].<br>Bits — Definition<br>00101b-00000b — Reserved<br>00110b — 400 MHz, 800 MT/s<br>01001b-00111b — Reserved<br>01010b — 533 MHz, 1066 MT/s<br>01101b-01011b — Reserved<br>01110b — 667 MHz, 1333 MT/s<br>10001b-01111b — Reserved<br>10010b — 800 MHz, 1600 MT/s<br>10101b-10011b — Reserved<br>10110b — 933 MHz, 1866 MT/s<br>11111b-10111b — Reserved |

### D18F2x[1,0]98 DRAM Controller Additional Data Offset Register

Reset: 8000_0000h.

Each DCT includes an array of registers that are used primarily to control DRAM-interface electrical parameters. Access to these registers is accomplished as follows:

- Reads:
  - Write the register number to D18F2x[1,0]98[DctOffset] with D18F2x[1,0]98[DctAccessWrite]=0.
  - Read the register contents from D18F2x[1,0]9C.

- Writes:
  - Write all 32 bits to the register data to D18F2x[1,0]9C (individual byte writes are not supported).
  - Write the register number to D18F2x[1,0]98[DctOffset] with D18F2x[1,0]98[DctAccessWrite]=1.

See 2.9.1 [DCT Configuration Registers] for general programming information about DCT configuration registers.

| Bits | Description |
|---|---|
| 31 | Reserved. |
| 30 | **DctAccessWrite: DRAM controller read/write select**. Read-write. 0=Read one of the F2x[1, 0]9C_x registers. 1=Write one of the F2x[1, 0]9C_x registers. |
| 29:0 | **DctOffset: DRAM controller offset**. Read-write. |

### D18F2x[1,0]9C DRAM Controller Additional Data Port

See D18F2x[1,0]98.

### D18F2x[1,0]9C_x0000_0000 DRAM Output Driver Compensation Control Register

Cold reset: 3033_3333h. BIOS: Table 23 -Table 24. See 2.9.3.4.6 [DRAM Address Timing and Output Driver Compensation Control].

| Bits | Description |
|---|---|
| 31 | Reserved. |
| 30:28 | **ProcOdt: processor on-die termination**. Read-write. This field specifies the resistance of the on-die termination resistors. This field is valid only when D18F2x[1,0]94[ProcOdtDis]=0.<br>Bits — Definition<br>000b — 240 ohms +/- 20%<br>001b — 120 ohms +/- 20%<br>010b — 80 ohms +/- 20%<br>011b — 60 ohms +/- 20%<br>111b-100b — Reserved. |
| 27:23 | Reserved. |
| 22:20 | **DqsDrvStren: DQS drive strength**. Read-write. This field specifies the drive strength of the DQS pins.<br>Bits — Definition<br>000b — 0.75x<br>001b — 1.0x<br>010b — 1.25x<br>011b — 1.5x<br>111b-100b — Reserved |
| 19 | Reserved. |

| | |
|---|---|
| 18:16 | **DataDrvStren: data drive strength**. Read-write. This field specifies the drive strength of the DRAM data pins.<br>Bits: Definition<br>000b: 0.75x<br>001b: 1.0x<br>010b: 1.25x<br>011b: 1.5x<br>111b-100b: Reserved |
| 15 | Reserved. |
| 14:12 | **ClkDrvStren: MEMCLK drive strength**. Read-write. This field specifies the drive strength of the MEMCLK pins.<br>Bits: Definition<br>000b: 1.0x<br>001b: 1.25x<br>010b: 1.5x<br>011b: 2.0x<br>111b-100b: Reserved |
| 11 | Reserved. |
| 10:8 | **AddrCmdDrvStren: address/command drive strength**. Read-write. This field specifies the drive strength of the address, RAS, CAS, WE, bank and parity pins.<br>Bits: Definition<br>000b: 1.0x<br>001b: 1.25x<br>010b: 1.5x<br>011b: 2.0x<br>111b-100b: Reserved |
| 7 | Reserved. |
| 6:4 | **CsOdtDrvStren: CS/ODT drive strength**. Read-write. This field specifies the drive strength of the CS and ODT pins.<br>Bits: Definition<br>000b: 1.0x<br>001b: 1.25x<br>010b: 1.5x<br>011b: 2.0x<br>111b-100b: Reserved |
| 3 | Reserved. |
| 2:0 | **CkeDrvStren: CKE drive strength**. Read-write. This field specifies the drive strength of the CKE pins.<br>Bits: Definition<br>000b: 1.0x<br>001b: 1.25x<br>010b: 1.5x<br>011b: 2.0x<br>111b-100b: Reserved |

**D18F2x[1,0]9C_x0000_0[1:0]0[2:1] DRAM Write Data Timing**

BIOS: See 2.9.3.7.4 [DQS Position Training]. These registers control the timing of write DQ with respect to

MEMCLK and allow transmit DQS to be centered in the data eye. The delay starts 1 UI before the rise edge of MEMCLK corresponding to the CAS-write-latency. The total delay for each byte is the sum of WrDatGross-Dly and WrDatFineDly.

**Table 75: Index addresses for** D18F2x[1,0]9C_x0000_0[1:0]0[2:1]

| D18F2x[1,0]98[31:16] | D18F2x[1,0]98[15:0] | | | |
|---|---|---|---|---|
| | 0102h | 0101h | 0002h | 0001h |
| 0000h | DIMM 1 Bytes 7-4 | DIMM 1 Bytes 3-0 | DIMM 0 Bytes 7-4 | DIMM 0 Bytes 3-0 |

**Table 76: Byte lane mapping for** D18F2x[1,0]9C_x0000_0[1:0]0[2:1]

| Register | Bits | | | |
|---|---|---|---|---|
| | 31:24 | 23:16 | 15:8 | 7:0 |
| D18F2x[1,0]9C_x0000_0[1:0]01 | Byte 3 | Byte 2 | Byte 1 | Byte 0 |
| D18F2x[1,0]9C_x0000_0[1:0]02 | Byte 7 | Byte 6 | Byte 5 | Byte4 |

| Bits | Description |
|---|---|
| 31:29 | **WrDatGrossDly: write data gross delay**. Read-write. Reset: 0. See: D18F2x[1,0]9C_x0000_0[1:0]0[2:1][7:5]. |
| 28:24 | **WrDatFineDly: write data fine delay**. Read-write. Cold reset: 0. See: D18F2x[1,0]9C_x0000_0[1:0]0[2:1][4:0]. |
| 23:21 | **WrDatGrossDly: write data gross delay**. Read-write. Reset: 0. See: D18F2x[1,0]9C_x0000_0[1:0]0[2:1][7:5]. |
| 20:16 | **WrDatFineDly: write data fine delay**. Read-write. Cold reset: 0. See: D18F2x[1,0]9C_x0000_0[1:0]0[2:1][4:0]. |
| 15:13 | **WrDatGrossDly: write data gross delay**. Read-write. Reset: 0. See: D18F2x[1,0]9C_x0000_0[1:0]0[2:1][7:5]. |
| 12:8 | **WrDatFineDly: write data fine delay**. Read-write. Cold reset: 0. See: D18F2x[1,0]9C_x0000_0[1:0]0[2:1][4:0]. |
| 7:5 | **WrDatGrossDly: write data gross delay**. Read-write. Reset: 0.<br>Bits Definition<br>100b-000b <WrDatGrossDly*0.5> MEMCLK delay<br>111b-101b Reserved<br>WrDatGrossDly must be programmed to ensure that<br>WrDatGrossDly - D18F2x[1,0]9C_x0000_00[44:30][WrDqsGrossDly] <= 0.5 MEMCLKs. |
| 4:0 | **WrDatFineDly: write data fine delay**. Read-write. Cold reset: 0.<br>Bits Definition<br>1_1111b-0_0000b <WrDatFineDly>/64 MEMCLK delay |

### D18F2x[1,0]9C_x0000_0004 DRAM Address/Command Timing Control Register

BIOS: Table 23 -Table 24. See 2.9.3.4.6 [DRAM Address Timing and Output Driver Compensation Control].

This register controls the timing of the address, command, chip select, ODT and clock enable pins with respect to MEMCLK. See the figure below. This register is used to adjust both the setup and hold time at the DIMM.

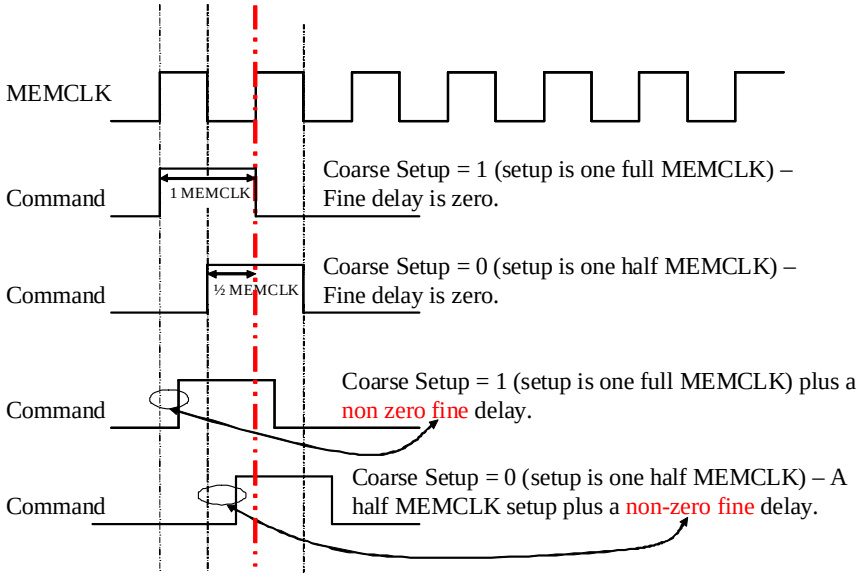


**Figure 14: Address/command timing at the processor pins**

2T timing is controlled by D18F2x[1,0]94[SlowAccessMode].

| Bits | Description |
|---|---|
| 31:22 | Reserved. |
| 21 | **AddrCmdSetup: address/command setup time**. Read-write. Reset: 0. This bit selects the default setup time for the address and command pins versus MEMCLK. 0=1/2 MEMCLK (1 1/2 MEMCLK for 2T timing). 1=MEMCLK (2 MEMCLKs for 2T timing). |
| 20:16 | **AddrCmdFineDelay: address/command fine delay**. Read-write. Cold reset: 0. This field specifies the time that the address and command pins are delayed from the default setup time. See: CkeFineDelay. |
| 15:14 | Reserved. |
| 13 | **CsOdtSetup: CS/ODT setup time**. Read-write. Reset: 0. This bit selects the default setup time for the CS and ODT pins versus MEMCLK. 0b=1/2 MEMCLK. 1b=1 MEMCLK. |
| 12:8 | **CsOdtFineDelay: CS/ODT fine delay**. Read-write. Cold reset: 0. This field specifies the time that the CS and ODT pins are delayed from the default setup time. See: CkeFineDelay. |
| 7:6 | Reserved. |
| 5 | **CkeSetup: CKE setup time**. Read-write. Reset: 0. This bit selects the default setup time for the CKE pins versus MEMCLK. 0b=1/2 MEMCLK. 1b=1 MEMCLK. |
| 4:0 | **CkeFineDelay: CKE fine delay**. Read-write. Cold reset: 0. This field specifies the time that the CKE pins are delayed from the default setup time.<br>Bits Definition<br>1_1111b-0_0000b <CkeFineDelay>/64 MEMCLK delay |

### D18F2x[1,0]9C_x0000_0[1:0]0[6:5] DRAM Read DQS Timing Control

Cold reset: 1E1E_1E1Eh. BIOS: See 2.9.3.7 [DRAM Training]. These registers delay the timing of read (input) DQS signals with respect to data.

**Table 77: Index addresses for** D18F2x[1,0]9C_x0000_0[1:0]0[6:5]

| D18F2x[1,0]98[31:16] | D18F2x[1,0]98[15:0] | | | |
|---|---|---|---|---|
| | 0106h | 0105h | 0006h | 0005h |
| 0000h | DIMM 1 Bytes 7-4 | DIMM 1 Bytes 3-0 | DIMM 0 Bytes 7-4 | DIMM 0 Bytes 3-0 |

**Table 78: Byte lane mapping for** D18F2x[1,0]9C_x0000_0[1:0]0[6:5]

| Register | Bits | | | |
|---|---|---|---|---|
| | 29:25 | 21:17 | 13:9 | 5:1 |
| D18F2x[1,0]9C_x0000_0[1:0]05 | Byte 3 | Byte 2 | Byte 1 | Byte 0 |
| D18F2x[1,0]9C_x0000_0[1:0]06 | Byte 7 | Byte 6 | Byte 5 | Byte4 |

| Bits | Description |
|---|---|
| 31:30 | Reserved. |
| 29:25 | **RdDqsTime: read DQS timing control**. Read-write. See: D18F2x[1,0]9C_x0000_0[1:0]0[6:5][5:1]. |
| 24:22 | Reserved. |
| 21:17 | **RdDqsTime: read DQS timing control**. Read-write. See: D18F2x[1,0]9C_x0000_0[1:0]0[6:5][5:1]. |
| 16:14 | Reserved. |
| 13:9 | **RdDqsTime: read DQS timing control**. Read-write. See: D18F2x[1,0]9C_x0000_0[1:0]0[6:5][5:1]. |
| 8:6 | Reserved. |
| 5:1 | **RdDqsTime: read DQS timing control**. Read-write.<br>Bits Definition<br>1_1111b-0_0000b <RdDqsTime>/64 MEMCLK delay |
| 0 | Reserved. |

### D18F2x[1,0]9C_x0000_0008 DRAM Phy Control Register

Cold reset: 0208_0000h.

**Table 79. DRAM Phy PLL Multiplier and Divide Ratio Values**

| MEMCLK Frequency(MHz) | PLL Multiplier | PLL Divide Ratio |
|---|---|---|
| 200 | 16 | 4 |
| 400 | 16 | 2 |
| 533 | 32 | 3 |
| 667 | 40 | 3 |
| 800 | 32 | 2 |
| 933 | 56 | 3 |

| Bits | Description |
|---|---|
| 31:28 | Reserved. |

| | |
|---|---|
| 27:24 | **PllDiv: phy PLL divider**. Read-only; updated-by-hardware. This field specifies the divide ratios that are encoded to the DDR phy PLL. Available encodings are as follows:<br>Encoding / Divide Ratio / Encoding / Divide Ratio<br>0000b 1 0110b 256<br>0001b 2 0111b PLL is disabled<br>0010b 4 1000b 3<br>0011b 8 1001b 6<br>0100b 16 1111b-1010b Reserved<br>0101b 128 |
| 23:22 | Reserved. |
| 21:15 | **PllMult: phy PLL multiplier**. Read-only; updated-by-hardware. This field specifies the multiplier values that are to be used for the DDR phy PLL.<br>Bits / Multiplier<br>7Fh-00h <PllMult> |
| 14 | Reserved. |
| 13 | **DqsRcvTrEn: DQS receiver training enable**. Read-write. BIOS: See 2.9.3.7.2. 1=Initiate hardware assisted read DQS receiver training. 0=Stop read DQS receiver training. This allows BIOS to reliably read the DQS receiver training data. |
| 12 | **WrLvOdtEn: write levelization ODT enabled**. Read-write. BIOS: See 2.9.3.7.1. 1=ODT specified by WrLvOdt is enabled during write levelization training. 0=ODT is disabled during write levelization training. |
| 11:8 | **WrLvOdt: write levelization ODT**. Read-write. BIOS: See 2.9.3.7.1. Specifies the state of the ODT pins when WrLvOdtEn=1. Bit[0] applies to ODT0; bit[1] applies to ODT1; etc. ODT numbers are as specified by [The DRAM CS Base Address Registers] D18F2x[1,0][4C:40]. For each bit, 1=ODT is enabled; 0=ODT is disabled. Tri-state enable for ODT is turned off by the phy while WrLvOdtEn=1. |
| 7:6 | **FenceTrSel: fence train select**. Read-write. BIOS: See 2.9.3.2.3. Specifies the flop to be used for phy based fence training. See PhyFenceTrEn.<br>Bits / Definition<br>00b PRE flop<br>01b RxDll flop<br>10b TxDll flop<br>11b TxPad flop |
| 5 | Reserved. |
| 4 | **TrDimmSel: training DIMM select**. Read-write. BIOS: See 2.9.3.7.1 and 2.9.3.7.2. Specifies which DIMM is to be trained. 0h=DIMM 0. 1h=DIMM 1. DIMM numbers are specified by Table 73. |
| 3 | **PhyFenceTrEn: phy fence training enable**. Read-write. BIOS: See 2.9.3.2.3. 1=Initiate phy based fence training. 0=Stop the phy based fence training engine. |
| 2 | Reserved. |
| 1 | Reserved. |
| 0 | **WrtLvTrEn: write levelization training enable**. Read-write. BIOS: See 2.9.3.7.1. 1=Initiate write levelization (tDQSS margining) training. 0=Stop driving DQS and exit write levelization training. |

### D18F2x[1,0]9C_x0000_000B DRAM Phy Status Register

Reset: 0000_0000h. RAZ; write.

| Bits | Description |
|---|---|
| 31 | **DynModeChange: dynamic mode change**. BIOS: See 2.9.3.8. 1=Phy enters the state specified by PhySelfRefreshMode. If D18F2x[1,0]94[DisDramInterface] is set, BIOS must also disable the phy for the channel by writing this register with DynModeChange=1 and PhySelfRefreshMode=1. |
| 30:24 | Reserved. |
| 23 | **PhySelfRefreshMode: phy self refresh mode**. 1=Enter self refresh mode. 0=Exit self refresh mode. See DynModeChange. |
| 22:0 | Reserved. |

### D18F2x[1,0]9C_x0000_000C DRAM Phy Miscellaneous Register

Cold reset: 4E73_0000h. BIOS: See 2.9.3.2.3 [Phy Fence Programming].

This register provides access to the DDR phy to control signal tri-state functionality. Based on the system configuration, BIOS may tri-state signals with associated chip selects that are unpopulated in an effort to conserve power. D18F2x9C_x0C controls the channel A pins while D18F2x19C_x0C controls channel B pins. See Table 73 for processor pin map. This register also provides access to the DDR phy fence logic used to adjust the phase relationship between the data FIFO and the data going to the pad.

| Bits | Description |
|---|---|
| 31 | Reserved. |
| 30:26 | **FenceThresholdTxDll: phy fence threshold transmit DLL**. Read-write. This field specifies the fence delay threshold value used for DQS receiver valid. This field is only used during DQS receiver enable training. See FenceThresholdTxPad. |
| 25:21 | **FenceThresholdRxDll: phy fence threshold DQS receiver enable**. Read-write. This field specifies the fence delay threshold value used for DQS receiver enable. See FenceThresholdTxPad. |
| 20:16 | **FenceThresholdTxPad: phy fence threshold transmit pad**. Read-write. This field specifies the fence delay threshold value used for write data, write DQS, Addr/Cmd, CS, ODT, and CKE.<br>Bits Definition<br>1_1111b-0_0000b <FenceThresholdTxPad>/64 MEMCLK delay |
| 15:14 | Reserved. |
| 13:12 | **CKETri: CKE tri-state**. Read-write. BIOS: See 2.9.3.8 and 2.9.3.9. 0=The CKE signals are not tri-stated. 1=Tri-state unconnected CKE signals from the processor. The bits CKETri[1:0] are mapped to packages as follows:<br>Bit Package pin name<br>[0] M[B,A]_CKE[0]<br>[1] M[B,A]_CKE[1] |

| | |
|---|---|
| 11:8 | **ODTTri: ODT tri-state**. Read-write. BIOS: See 2.9.3.9. 0=The ODT signals are not tri-stated unless directed by the DCT. 1=Tri-state unconnected ODT signals from the processor. The bits ODTTri[3:0] are mapped to packages as follows:<br>Bit Package pin name<br>[0] M[B,A]_ODT[0].<br>[1] M[B,A]_ODT[1].<br>[2] M[B,A]_ODT[2] (FM1 package only).<br>[3] M[B,A]_ODT[3] (FM1 package only). |
| 7:0 | **ChipSelTri: chip select tri-state**. Read-write. BIOS: See 2.9.3.9. 0=The chip select signals are not tri-stated unless directed to by the DCT. 1=Tri-state unpopulated chip selects when motherboard termination is available. The bits ChipSelTri[7:0] are mapped to packages as follows:<br>Bit Package pin name<br>[0] M[B,A]_CS_L[0].<br>[1] M[B,A]_CS_L[1].<br>[2] M[B,A]_CS_L[2] (FM1 package only).<br>[3] M[B,A]_CS_L[3] (FM1 package only).<br>[7:4] Unused pin. |

### D18F2x[1,0]9C_x0000_000D DRAM Phy DLL Control Register

Cold reset: 0000_0000h. This register defines programmable options for the phy's DLLs for power savings. There are two identical sets of configuration registers: one for the transmit DLLs (those running off of the phy's internal PCLK which is running at rate of 2*MEMCLK) and receive DLLs (those running off of the DQS from the DIMMs).

| Bits | Description |
|---|---|
| 31:26 | Reserved. |
| 25:24 | **RxDLLWakeupTime: receive DLL wakeup time**. Read-write. BIOS: See 2.9.3.9.This field specifies the number of PCLKs that the DLL standby signal must deassert prior to a DLL relock event or before read traffic is sent to the receive DLLs. |
| 23 | Reserved. |
| 22:20 | **RxCPUpdPeriod: receive charge pump period**. Read-write. BIOS: See 2.9.3.9. This field specifies the number of DLL relocks required to keep the receive DLLs locked for the period where there is no read traffic. |
| 19:16 | **RxMaxDurDllNoLock: receive maximum duration DLL no lock**. Read-write. Read-write. BIOS: See 2.9.3.5. This field specifies the number of PCLK cycles that occur before the phy DLLs relock. A DLL relock occurs every 2^RxMaxDurDllNoLock if there are no reads during the period. 0=DLL power saving disabled. |
| 15:10 | Reserved. |
| 9:8 | **TxDLLWakeupTime: transmit DLL wakeup time**. Read-write. BIOS: See 2.9.3.9. This field specifies the number of PCLK's that the DLL standby signal must deassert prior to a DLL relock event or before write traffic is sent to transmit DLLs. |
| 7 | Reserved. |

| | |
|---|---|
| 6:4 | **TxCPUpdPeriod: transmit charge pump period**. Read-write. BIOS: See 2.9.3.9. This specifies the number of DLL relocks required to keep the TxDLLs locked for the period where there is no write traffic. |
| 3:0 | **TxMaxDurDllNoLock: transmit maximum duration DLL no lock**. Read-write. BIOS: See 2.9.3.5. This field specifies the number of PCLK cycles that occur before the phy DLLs relock. A DLL relock occurs every 2^TxMaxDurDllNoLock if there are no writes during the period. 0=DLL power saving disabled. |

### D18F2x[1,0]9C_x0000_00[24:10] DRAM DQS Receiver Enable Timing Control

BIOS: See 2.9.3.7 [DRAM Training]. Each of these registers control the timing of the receiver enable from the start of the read preamble with respect to MEMCLK. Each control includes a gross timing field and a fine timing field, the sum of which is the total delay.

**Table 80: Index addresses for** D18F2x[1,0]9C_x0000_00[24:10]

| D18F2x[1,0]98[31:4] | D18F2x[1,0]98[3:0] | | | |
|---|---|---|---|---|
| | 4h | 3h | 1h | 0h |
| 000_0001h | DIMM 1 Bytes 3-2 | DIMM 1 Bytes 1-0 | DIMM 0 Bytes 3-2 | DIMM 0 Bytes 1-0 |
| 000_0002h | DIMM 1 Bytes 7-6 | DIMM 1 Bytes 5-4 | DIMM 0 Bytes 7-6 | DIMM 0 Bytes 5-4 |

**Table 81: Byte lane mapping for** D18F2x[1,0]9C_x0000_00[24:10]

| Register | Bits | |
|---|---|---|
| | 24:16 | 8:0 |
| D18F2x[1,0]9C_x0000_001[0,3] | Byte 1 | Byte 0 |
| D18F2x[1,0]9C_x0000_001[1,4] | Byte 3 | Byte 2 |
| D18F2x[1,0]9C_x0000_002[0,3] | Byte 5 | Byte 4 |
| D18F2x[1,0]9C_x0000_002[1,4] | Byte 7 | Byte 6 |

| Bits | Description |
|---|---|
| 31:25 | Reserved. |
| 24:21 | **DqsRcvEnGrossDelay: DQS receiver enable gross delay**. Read-write. Reset: 1. See: D18F2x[1,0]9C_x0000_00[24:10][8:5]. |
| 20:16 | **DqsRcvEnFineDelay: DQS receiver enable fine delay**. Read-write. Cold reset: 0. See: D18F2x[1,0]9C_x0000_00[24:10][4:0]. |
| 15:9 | Reserved. |
| 8:5 | **DqsRcvEnGrossDelay: DQS receiver enable gross delay**. Read-write. Reset: 1.<br>Bits Description<br>Ch-0h <DqsRcvEnGrossDelay*0.5> MEMCLK delay<br>Fh-Dh Reserved |
| 4:0 | **DqsRcvEnFineDelay: DQS receiver enable fine delay**. Read-write. Cold reset: 0.<br>Bits Definition<br>1_1111b-0_0000b <DqsRcvEnFineDelay>/64 MEMCLK delay |

### D18F2x[1,0]9C_x0000_00[44:30] DRAM DQS Write Timing Control

BIOS: See 2.9.3.7 [DRAM Training]. These registers control the timing of write DQS with respect to MEMCLK. The delay starts at the rise edge of MEMCLK corresponding to the CAS-write-latency. The total delay for each byte is the sum of WrDqsGrossDly and WrDqsFineDly.

**Table 82: Index addresses for** D18F2x[1,0]9C_x0000_00[44:30]

| D18F2x[1,0]98[31:4] | D18F2x[1,0]98[3:0] | | | |
|---|---|---|---|---|
| | 4h | 3h | 1h | 0h |
| 000_0003h | DIMM 1 Bytes 3-2 | DIMM 1 Bytes 1-0 | DIMM 0 Bytes 3-2 | DIMM 0 Bytes 1-0 |
| 000_0004h | DIMM 1 Bytes 7-6 | DIMM 1 Bytes 5-4 | DIMM 0 Bytes 7-6 | DIMM 0 Bytes 5-4 |

**Table 83: Byte lane mapping for** D18F2x[1,0]9C_x0000_00[44:30]

| Register | Bits | |
|---|---|---|
| | 23:16 | 7:0 |
| D18F2x[1,0]9C_x0000_003[0,3] | Byte 1 | Byte 0 |
| D18F2x[1,0]9C_x0000_003[1,4] | Byte 3 | Byte 2 |
| D18F2x[1,0]9C_x0000_004[0,3] | Byte 5 | Byte 4 |
| D18F2x[1,0]9C_x0000_004[1,4] | Byte 7 | Byte 6 |

| Bits | Description |
|---|---|
| 31:24 | Reserved. |
| 23:21 | **WrDqsGrossDly: DQS write gross delay**. Read-write. Reset: 0. See: D18F2x[1,0]9C_x0000_00[44:30][7:5]. |
| 20:16 | **WrDqsFineDly: DQS write fine delay**. Read-write. Cold reset: 0. See: D18F2x[1,0]9C_x0000_00[44:30][4:0]. |
| 15:8 | Reserved. |
| 7:5 | **WrDqsGrossDly: DQS write gross delay**. Read-write. Reset: 0.<br>Bits Definition<br>100b-000b <WrDqsGrossDly*0.5> MEMCLK delay<br>111b-101b Reserved |
| 4:0 | **WrDqsFineDly: DQS write fine delay**. Read-write. Cold reset: 0.<br>Bits Definition<br>1_1111b-0_0000b <WrDqsFineDly>/64 MEMCLK delay |

### D18F2x[1,0]9C_x0000_00[51:50] DRAM Phase Recovery Control

BIOS: See 2.9.3.7 [DRAM Training]. These registers are used for hardware assisted DRAM training. Writes to these registers seed the phase recovery engine prior to training. Reads from the registers indicate how much the phase recovery engine has advanced to align the MEMCLK and DQS edges and is under hardware control. The total delay for each byte is the sum of PhRecGrossDlyByte and PhRecFineDlyByte, ranging from 0 to 1 and 63/64 MEMCLKs.

**Table 84: Index addresses for** D18F2x[1,0]9C_x0000_00[51:50]

| D18F2x[1,0]98[31:8] | D18F2x[1,0]98[7:0] | |
|---|---|---|
| | 51h | 50h |
| 00_0000h | Bytes 7-4 | Bytes 3-0 |

**Table 85: Byte lane mapping for** D18F2x[1,0]9C_x0000_00[51:50]

| Register | Bits | | | |
|---|---|---|---|---|
| | 30:24 | 22:16 | 14:8 | 6:0 |
| D18F2x[1,0]9C_x0000_0050 | Byte 3 | Byte 2 | Byte 1 | Byte 0 |
| D18F2x[1,0]9C_x0000_0051 | Byte 7 | Byte 6 | Byte 5 | Byte 4 |

| Bits | Description |
|---|---|
| 31 | Reserved. |
| 30:29 | **PhRecGrossDly: phase recovery gross delay**. Read-write; updated-by-hardware. Reset: X. See: D18F2x[1,0]9C_x0000_00[51:50][6:5]. |
| 28:24 | **PhRecFineDly: phase recovery fine delay**. Read-write; updated-by-hardware. Reset: X. See: D18F2x[1,0]9C_x0000_00[51:50][4:0]. |
| 23 | Reserved. |
| 22:21 | **PhRecGrossDly: phase recovery gross delay byte**. Read-write; updated-by-hardware. Reset: X. See: D18F2x[1,0]9C_x0000_00[51:50][6:5]. |
| 20:16 | **PhRecFineDly: phase recovery fine delay byte**. Read-write; updated-by-hardware. Reset: X. See: D18F2x[1,0]9C_x0000_00[51:50][4:0]. |
| 15 | Reserved. |
| 14:13 | **PhRecGrossDly: phase recovery gross delay**. Read-write; updated-by-hardware. Reset: X. See: D18F2x[1,0]9C_x0000_00[51:50][6:5]. |
| 12:8 | **PhRecFineDly: phase recovery fine delay**. Read-write; updated-by-hardware. Reset: X. See: D18F2x[1,0]9C_x0000_00[51:50][4:0]. |
| 7 | Reserved. |
| 6:5 | **PhRecGrossDly: phase recovery gross delay**. Read-write; updated-by-hardware. Reset: X.<br>Bits Definition<br>11b-00b <PhRecGrossDly*0.5> MEMCLKs |
| 4:0 | **PhRecFineDly: phase recovery fine delay**. Read-write; updated-by-hardware. Reset: X.<br>Bits Definition<br>1_1111b-0_0000b <PhRecFineDly>/64 MEMCLK delay |

### D18F2x[1,0]9C_x0D0F_0[F,7:0]02 Data Byte Transmit PreDriver Calibration

Cold reset: xxxx_xxxxh. BIOS: See 2.9.3.2.4.

**Table 86: Index addresses for** D18F2x[1,0]9C_x0D0F_0[F,7:0]02

| D18F2x[1,0]98[31:16] | D18F2x[1,0]98[15:0] | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 0702h | 0602h | 0502h | 0402h | 0302h | 0202h | 0102h | 0002h |
| 0D0Fh | Byte 7 | Byte 6 | Byte 5 | Byte 4 | Byte 3 | Byte 2 | Byte 1 | Byte 0 |

**Table 87: Broadcast write index address for** D18F2x[1,0]9C_x0D0F_0[F,7:0]02

| D18F2x[1,0]98[31:16] | D18F2x[1,0]98[15:0] |
|---|---|
| | 0F02h |
| 0D0Fh | D18F2x[1,0]9C_x0D0F_0[7:0]02 |

Table 88: **Valid values for** D18F2x[1,0]9C_x0D0F_0[F,7:0]02[TxPreP, TxPreN]

| Bits | Description |
|---|---|
| 0h | Slew Rate 0 (slowest) |
| 8h-1h | Reserved |
| 9h | Slew Rate 1 |
| 11h-Ah | Reserved |
| 12h | Slew Rate 2 |
| 1Ah-13h | Reserved |
| 1Bh | Slew Rate 3 |
| 23h-1Ch | Reserved |
| 24h | Slew Rate 4 |
| 2Ch-25h | Reserved |
| 2Dh | Slew Rate 5 |
| 35h-2Eh | Reserved |
| 36h | Slew Rate 6 |
| 3Eh-37h | Reserved |
| 3Fh | Slew Rate 7 (fastest) |

| Bits | Description |
|---|---|
| 31:16 | Reserved. |
| 15 | **ValidTxAndPre: predriver calibration code valid**. Read-write; cleared-when-done. 1=Predriver calibration codes are copied from this register and D18F2x[1,0]9C_x0D0F_0[F,7:0]0[A,6] into the associated transmit pad. |
| 14:12 | Reserved. |
| 11:6 | **TxPreP: PMOS predriver calibration code**. Read-write. Specifies the rising edge slew rate of the transmit pad. See: Table 88 [Valid values for D18F2x[1,0]9C_x0D0F_0[F,7:0]02[TxPreP, TxPreN]]. After updating this value, BIOS must program ValidTxAndPre=1 for the change to take effect. |
| 5:0 | **TxPreN: NMOS predriver calibration code**. Read-write. Specifies the falling edge slew rate of the transmit pad. See: Table 88 [Valid values for D18F2x[1,0]9C_x0D0F_0[F,7:0]02[TxPreP, TxPreN]]. After updating this value, BIOS must program ValidTxAndPre=1 for the change to take effect. |

### D18F2x[1,0]9C_x0D0F_0[F,7:0]0[A,6] Data Byte Transmit PreDriver Calibration 2

Cold reset: xxxx_xxxxh. BIOS: See 2.9.3.2.4.

**Table 89: Index addresses for D18F2x[1,0]9C_x0D0F_0[F,7:0]0[A,6] pad group 1**

| D18F2x[1,0]98[31:16] | D18F2x[1,0]98[15:0] | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 0706h | 0606h | 0506h | 0406h | 0306h | 0206h | 0106h | 0006h |
| 0D0Fh | Byte 7 | Byte 6 | Byte 5 | Byte 4 | Byte 3 | Byte 2 | Byte 1 | Byte 0 |

**Table 90: Index addresses for D18F2x[1,0]9C_x0D0F_0[F,7:0]0[A,6] pad group 2**

| D18F2x[1,0]98[31:16] | D18F2x[1,0]98[15:0] | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 070Ah | 060Ah | 050Ah | 040Ah | 030Ah | 020Ah | 010Ah | 000Ah |
| 0D0Fh | Byte 7 | Byte 6 | Byte 5 | Byte 4 | Byte 3 | Byte 2 | Byte 1 | Byte 0 |

**Table 91: Broadcast write index address for D18F2x[1,0]9C_x0D0F_0[F,7:0]0[A,6]**

| D18F2x[1,0]98[31:16] | D18F2x[1,0]98[15:0] | |
|---|---|---|
| | 0F0Ah | 0F06h |
| 0D0Fh | D18F2x[1,0]9C_x0D0F_0[7:0]0A | D18F2x[1,0]9C_x0D0F_0[7:0]06 |

| Bits | Description |
|---|---|
| 31:12 | Reserved. |
| 11:6 | **TxPreP: PMOS predriver calibration code**. Read-write. Specifies the rising edge slew rate of the transmit pad. See: Table 88 [Valid values for D18F2x[1,0]9C_x0D0F_0[F,7:0]02[TxPreP, TxPreN]]. After updating this value, BIOS must program D18F2x[1,0]9C_x0D0F_0[F,7:0]02[ValidTxAnd-Pre]=1 for the change to take effect. |
| 5:0 | **TxPreN: NMOS predriver calibration code**. Read-write. Specifies the falling edge slew rate of the transmit pad. See: Table 88 [Valid values for D18F2x[1,0]9C_x0D0F_0[F,7:0]02[TxPreP, TxPreN]]. After updating this value, BIOS must program D18F2x[1,0]9C_x0D0F_0[F,7:0]02[ValidTxAnd-Pre]=1 for the change to take effect. |

### D18F2x[1,0]9C_x0D0F_0[F,7:0]0F Data Byte DLL Clock Enable

Cold reset: 0000_0013h.

**Table 92: Index addresses for D18F2x[1,0]9C_x0D0F_0[F,7:0]0F**

| D18F2x[1,0]98[31:16] | D18F2x[1,0]98[15:0] | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 070Fh | 060Fh | 050Fh | 040Fh | 030Fh | 020Fh | 010Fh | 000Fh |
| 0D0Fh | Byte 7 | Byte 6 | Byte 5 | Byte 4 | Byte 3 | Byte 2 | Byte 1 | Byte 0 |

**Table 93: Broadcast write index address for D18F2x[1,0]9C_x0D0F_0[F,7:0]0F**

| D18F2x[1,0]98[31:16] | D18F2x[1,0]98[15:0] |
|---|---|
| | 0F0Fh |
| 0D0Fh | D18F2x[1,0]9C_x0D0F_0[7:0]0F |

| Bits | Description |
|---|---|
| 31:15 | Reserved. |
| 14:12 | **AlwaysEnDllClks: always enable DLL clocks**. Read-write. BIOS: See 2.9.3.2.3.0=DLL clocks are turned off during periods of inactivity. 1=DLL clocks remain on during inactivity. Prior to programming AlwaysEnDllClks to a value other than 000b, D18F2x[1,0]9C_x0000_000D[RxMaxDurDllNoLock, TxMaxDurDllNoLock] must both be programmed to 0000b. The bits AlwaysEnDllClks[2:0] are mapped to DLLs as follows:<br>Bit Definition<br>[2] TxDqs DLL<br>[1] TxDq DLL<br>[0] RxEn DLL |
| 11:0 | Reserved. |

### D18F2x[1,0]9C_x0D0F_0[F,7:0]10 Data Byte DLL Power Management

**Table 94: Index addresses for** D18F2x[1,0]9C_x0D0F_0[F,7:0]10

| D18F2x[1,0]98[31:16] | D18F2x[1,0]98[15:0] | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 0710h | 0610h | 0510h | 0410h | 0310h | 0210h | 0110h | 0010h |
| 0D0Fh | Byte 7 | Byte 6 | Byte 5 | Byte 4 | Byte 3 | Byte 2 | Byte 1 | Byte 0 |

**Table 95: Broadcast write index address for** D18F2x[1,0]9C_x0D0F_0[F,7:0]10

| D18F2x[1,0]98[31:16] | D18F2x[1,0]98[15:0] |
|---|---|
| | 0F10h |
| 0D0Fh | D18F2x[1,0]9C_x0D0F_0[7:0]10 |

| Bits | Description |
|---|---|
| 31:13 | Reserved. |
| 12 | **EnRxPadStandby: enable receiver pad standby**. Read-write. Cold reset: 0. BIOS: See 2.9.3.5.1=Phy will enable Receiver Standby Mode when it is not receiving data to save power. |
| 11:0 | Reserved. |

### D18F2x[1,0]9C_x0D0F_0[F,7:0]13 Data Byte DLL Configuration

**Table 96: Index addresses for** D18F2x[1,0]9C_x0D0F_0[F,7:0]13

| D18F2x[1,0]98[31:16] | D18F2x[1,0]98[15:0] | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 0713h | 0613h | 0513h | 0413h | 0313h | 0213h | 0113h | 0013h |
| 0D0Fh | Byte 7 | Byte 6 | Byte 5 | Byte 4 | Byte 3 | Byte 2 | Byte 1 | Byte 0 |

**Table 97: Broadcast write index address for** D18F2x[1,0]9C_x0D0F_0[F,7:0]13

| D18F2x[1,0]98[31:16] | D18F2x[1,0]98[15:0] |
|---|---|
| | 0F13h |
| 0D0Fh | D18F2x[1,0]9C_x0D0F_0[7:0]13 |

| Bits | Description |
|---|---|
| 31:15 | Reserved. |
| 14 | **ProcOdtAdv: ProcOdt advance**. Read-write. Cold reset: 1. BIOS: See 2.9.3.2.2. 1=Start the digital control for the ProcOdt (receive termination) 1 PCLK early. |
| 13:8 | Reserved. |
| 7 | **RxDqsUDllPowerDown: receive DQS upper DLL power down**. Read-write. Cold reset: 0. BIOS: See 2.9.3.9. 1=Power down the upper receiver DQS DLL. BIOS should set this bit when x4 DIMMs are not present. |
| 6:2 | Reserved. |
| 1 | **DllDisEarlyU: DLL disable early upper**. Read-write. Cold reset: 0. BIOS: See 2.9.3.9. 1=Disable upper receiver DQS DLL early timing for power savings. |
| 0 | **DllDisEarlyL: DLL disable early lower**. Read-write. Cold reset: 0. BIOS: See 2.9.3.9. 1=Disable lower receiver DQS DLL early timing for power savings. |

### D18F2x[1,0]9C_x0D0F_0[F,7:0]1F Data Byte Receiver Configuration

Cold reset: 0000_2002h.

**Table 98: Index addresses for** D18F2x[1,0]9C_x0D0F_0[F,7:0]1F

| D18F2x[1,0]98[31:16] | D18F2x[1,0]98[15:0] | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 071Fh | 061Fh | 051Fh | 041Fh | 031Fh | 021Fh | 011Fh | 001Fh |
| 0D0Fh | Byte 7 | Byte 6 | Byte 5 | Byte 4 | Byte 3 | Byte 2 | Byte 1 | Byte 0 |

**Table 99: Broadcast write index address for** D18F2x[1,0]9C_x0D0F_0[F,7:0]1F

| D18F2x[1,0]98[31:16] | D18F2x[1,0]98[15:0] |
|---|---|
| | 0F1Fh |
| 0D0Fh | D18F2x[1,0]9C_x0D0F_0[7:0]1F |

| Bits | Description |
|---|---|
| 31:5 | Reserved. |

| | |
|---|---|
| 4:3 | **RxVioLvl: receiver voltage level**. Read-write. BIOS: See 2.9.3.2.1. This field specifies the VDDIO voltage level.<br>Bits Definition<br>00b 1.5V<br>01b 1.35V<br>10b Reserved<br>11b Reserved |
| 2:0 | Reserved. |

### D18F2x[1,0]9C_x0D0F_0[F,7:0]30 Data Byte DLL Configuration and PowerDown

**Table 100: Index addresses for** D18F2x[1,0]9C_x0D0F_0[F,7:0]30

| D18F2x[1,0]98[31:16] | D18F2x[1,0]98[15:0] | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 0730h | 0630h | 0530h | 0430h | 0330h | 0230h | 0130h | 0030h |
| 0D0Fh | Byte 7 | Byte 6 | Byte 5 | Byte 4 | Byte 3 | Byte 2 | Byte 1 | Byte 0 |

**Table 101: Broadcast write index address for** D18F2x[1,0]9C_x0D0F_0[F,7:0]30

| D18F2x[1,0]98[31:16] | D18F2x[1,0]98[15:0] |
|---|---|
| | 0F30h |
| 0D0Fh | D18F2x[1,0]9C_x0D0F_0[7:0]30 |

| Bits | Description |
|---|---|
| 31:9 | Reserved. |
| 8 | **BlockRxDqsLock: block receive DQS lock**. Read-write. Cold reset: 0. BIOS: 2.9.3.7.3. Specifies how the receive DLLs lock. 1=Lock on PCLK. 0=Lock on both PCLK and the received DQS. |
| 7:6 | Reserved. |
| 5 | **PchgPdTxCClkGateDis: precharge power down TxCCLK gate disable**. Read-write. Cold reset: 0. BIOS: 0. 1=Disable gating of upstream TxCCLK during precharge power down. |
| 4:0 | Reserved. |

### D18F2x[1,0]9C_x0D0F_0[F,7:0]31 Data Byte Fence2 Threshold

BIOS: See 2.9.3.2.3 [Phy Fence Programming].

**Table 102: Index addresses for** D18F2x[1,0]9C_x0D0F_0[F,7:0]31

| D18F2x[1,0]98[31:16] | D18F2x[1,0]98[15:0] | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 0731h | 0631h | 0531h | 0431h | 0331h | 0231h | 0131h | 0031h |
| 0D0Fh | Byte 7 | Byte 6 | Byte 5 | Byte 4 | Byte 3 | Byte 2 | Byte 1 | Byte 0 |

**Table 103: Broadcast write index address for** D18F2x[1,0]9C_x0D0F_0[F,7:0]31

| D18F2x[1,0]98[31:16] | D18F2x[1,0]98[15:0] |
|---|---|
| | 0F31h |
| 0D0Fh | D18F2x[1,0]9C_x0D0F_0[7:0]31 |

| Bits | Description |
|---|---|
| 31:15 | Reserved. |
| 14 | **Fence2EnableRxDll: phy fence2 enable receive DLL**. Read-write. Cold reset: 0. 1=Enable the use of Fence2ThresholdRxDll for DQS receiver enable fence threshold. 0=Use D18F2x[1,0]9C_x0000_000C[FenceThresholdRxDll]. |
| 13:10 | **Fence2ThresholdRxDll: phy fence2 threshold DQS receiver enable**. Read-write. Cold reset: 0. If Fence2EnableRxDll=1**,** this field specifies the fence delay threshold value used for DQS receiver enable. See Fence2ThresholdTxPad. |
| 9 | **Fence2EnableTxDll: phy fence2 enable transmit DLL**. Read-write. Cold reset: 0. 1=Enable the use of Fence2ThresholdTxDll for transmit DLL fence threshold. 0=Use D18F2x[1,0]9C_x0000_000C[FenceThresholdTxDll]. |
| 8:5 | **Fence2ThresholdTxDll: phy fence2 threshold transmit DLL**. Read-write. Cold reset: 0. If Fence2EnableTxDll=1**,** this field specifies the fence delay threshold value used for DQS receiver valid. This field is only used during DQS receiver enable training. See Fence2ThresholdTxPad. |
| 4 | **Fence2EnableTxPad: fence2 enable transmit pad**. Read-write. Cold reset: 0. 1=Enable the use of Fence2ThresholdTxPad for transmit pad fence threshold. 0=Use D18F2x[1,0]9C_x0000_000C[FenceThresholdTxPad]. |
| 3:0 | **Fence2ThresholdTxPad: phy fence2 threshold transmit pad**. Read-write. Cold reset: 0. If Fence2EnableTxPad=1**,** this field specifies the fence delay threshold value used for write data and write DQS.<br>Bits Definition<br>Fh-0h <Fence2ThresholdTxPad>/64 MEMCLK delay |

### D18F2x[1,0]9C_x0D0F_2[1:0]02 Clock Transmit PreDriver Calibration

Cold reset: xxxx_xxxxh. BIOS: See 2.9.3.2.4.

**Table 104: Index address mapping for** D18F2x[1,0]9C_x0D0F_2[1:0]02

| D18F2x[1,0]98[31:0] | Function |
|---|---|
| 0D0F_2002h | Clock 0 Pad Group 0 |
| 0D0F_2102h | Clock 1 Pad Group 0 |

| Bits | Description |
|---|---|
| 31:16 | Reserved. |
| 15 | **ValidTxAndPre: predriver calibration code valid**. Read-write; cleared-when-done. 1=Predriver calibration codes are copied from this register into the associated transmit pad. |
| 14:12 | Reserved. |

| | |
|---|---|
| 11:6 | **TxPreP: PMOS predriver calibration code**. Read-write. Specifies the rising edge slew rate of the transmit pad. See: Table 88 [Valid values for D18F2x[1,0]9C_x0D0F_0[F,7:0]02[TxPreP, TxPreN]]. After updating this value, BIOS must program ValidTxAndPre=1 for the change to take effect. |
| 5:0 | **TxPreN: NMOS predriver calibration code**. Read-write. Specifies the falling edge slew rate of the transmit pad. See: Table 88 [Valid values for D18F2x[1,0]9C_x0D0F_0[F,7:0]02[TxPreP, TxPreN]]. After updating this value, BIOS must program ValidTxAndPre=1 for the change to take effect. |

### D18F2x[1,0]9C_x0D0F_[C,8,2][F,1:0]1F Receiver Configuration

Cold reset: 0000_2000h.

**Table 105: Index address mapping for** D18F2x[1,0]9C_x0D0F_[C,8,2][F,1:0]1F

| Index | Function | Index | Function |
|---|---|---|---|
| 0D0F_201Fh | Clock 0 | 0D0F_811Fh | Cmd/Addr 1 |
| 0D0F_211Fh | Clock 1 | 0D0F_C01Fh | Address |
| 0D0F_801Fh | Cmd/Addr 0 | - | - |

**Table 106: Broadcast write index address for** D18F2x[1,0]9C_x0D0F_[C,8,2][F,1:0]1F

| D18F2x[1,0]98[31:16] | D18F2x[1,0]98[15:0] | |
|---|---|---|
| | 8F1Fh | 2F1Fh |
| 0D0Fh | D18F2x[1,0]9C_x0D0F_8[1:0]1F | D18F2x[1,0]9C_x0D0F_2[1:0]1F |

| Bits | Description |
|---|---|
| 31:5 | Reserved. |
| 4:3 | **RxVioLvl: receiver voltage level**. Read-write. BIOS: See 2.9.3.2.1. This field specifies the VDDIO voltage level.<br>Bits Definition<br>00b 1.5V<br>01b 1.35V<br>10b Reserved<br>11b Reserved |
| 2:0 | Reserved. |

### D18F2x[1,0]9C_x0D0F_2[1:0]20 Clock DLL Delay and Configuration

Cold reset: 0000_8000h. BIOS: See 2.9.3.7.1. These registers control the timing of MEMCLK with respect to the phy internal clock.

**Table 107: Index address mapping for** D18F2x[1,0]9C_x0D0F_2[1:0]20

| D18F2x[1,0]98[31:0] | Function |
|---|---|
| 0D0F_2020h | Clock 0 |
| 0D0F_2120h | Clock 1 |

| Bits | Description |
|---|---|
| 31:15 | Reserved. |
| 14 | **DllNukeLoad: dll nuke load**. Read-write; updated-by-hardware. BIOS: See 2.9.3.7.1. 1=Clear the DLL delay line and internal state machines. 0=Latch the DLL fine delay input value into the DLL. This bit must be set and then cleared anytime a new DLL delay value is written. It can be set simultaneously with the new DLL delay. |
| 13:8 | Reserved. |
| 7 | **FenceBit: Fence bit**. Read-write. BIOS: See 2.9.3.7.1. 1=MEMCLK Tx pad input delayed by 0.5 UI. 0=MEMCLK Tx pad input not delayed. |
| 6:5 | Reserved. |
| 4:0 | **ClkFineDly: MEMCLK fine delay**. Read-write. BIOS: See 2.9.3.7.1.<br>Bits Definition<br>1_1111b-0_0000b <ClkFineDly>/64 MEMCLK delay |

### D18F2x[1,0]9C_x0D0F_2[1:0]30 Clock Configuration and Power Down

**Table 108: Index address mapping for** D18F2x[1,0]9C_x0D0F_2[1:0]30

| D18F2x[1,0]98[31:0] | Function | MemClkDis mapping |
|---|---|---|
| 0D0F_2030h | Clock 0 | D18F2x[1,0]88[MemClkDis[1:0]] |
| 0D0F_2130h | Clock 1 | D18F2x[1,0]88[MemClkDis[3:2]] |

**Table 109: MemClkDis mapping for** D18F2x[1,0]9C_x0D0F_2[1:0]30

| D18F2x[1,0]98[11:8] | |
|---|---|
| 0h | D18F2x[1,0]88[MemClkDis[1:0]] |
| 1h | D18F2x[1,0]88[MemClkDis[3:2]] |

| Bits | Description |
|---|---|
| 31:5 | Reserved. |
| 4 | **PwrDn: power down**. Read-write. Cold reset: 0. BIOS: See 2.9.3.9. 1=Turn off DLL circuitry. BIOS should set this bit if both lanes of the clock lane pair are not supported by the package. |
| 3:0 | Reserved. |

### D18F2x[1,0]9C_x0D0F_[C,8,2][F,1:0]31 Fence2 Threshold

BIOS: See 2.9.3.2.3 [Phy Fence Programming].

**Table 110: Index address mapping for** D18F2x[1,0]9C_x0D0F_[C,8,2][F,1:0]31

| Index | Function | Index | Function |
|---|---|---|---|
| 0D0F_2031h | Clock 0 | 0D0F_8131h | Cmd/Addr 1 |
| 0D0F_2131h | Clock 1 | 0D0F_C031h | Address |
| 0D0F_8031h | Cmd/Addr 0 | - | - |

**Table 111: Broadcast write index address for** D18F2x[1,0]9C_x0D0F_[C,8,2][F,1:0]31

| D18F2x[1,0]98[31:16] | D18F2x[1,0]98[15:0] | |
|---|---|---|
| | 8F31h | 2F31h |
| 0D0Fh | D18F2x[1,0]9C_x0D0F_8[1:0]31 | D18F2x[1,0]9C_x0D0F_2[1:0]31 |

| Bits | Description |
|---|---|
| 31:5 | Reserved. |
| 4 | **Fence2EnableTxPad: fence2 enable transmit pad**. Read-write. Cold reset: 0. 1=Enable the use of Fence2ThresholdTxPad for transmit pad fence threshold. 0=Use D18F2x[1,0]9C_x0000_000C[FenceThresholdTxPad]. |
| 3:0 | **Fence2ThresholdTxPad: phy fence2 threshold transmit pad**. Read-write. Cold reset: 0. If Fence2EnableTxPad=1**,** this field specifies the fence delay threshold value used for CLK, Addr/Cmd, CS, ODT, and CKE.<br>Bits Definition<br>Fh-0h <Fence2ThresholdTxPad>/64 MEMCLK delay |

### D18F2x[1,0]9C_x0D0F_4009 Cmp Receiver Configuration

Cold reset: 0000_2000h.

| Bits | Description |
|---|---|
| 31:16 | Reserved. |
| 15:14 | **CmpVioLvl: receiver voltage level**. Read-write. BIOS: See 2.9.3.2.1. This field specifies the VDDIO voltage level. Setting this field in DCT0 adjusts the VDDIO voltage level for DCT0 and DCT1. Setting this field in DCT1 has no effect.<br>Bits Definition<br>00b 1.5V<br>01b 1.35V<br>10b Reserved<br>11b Reserved |
| 13:4 | Reserved. |
| 3:2 | **ComparatorAdjust: comparator adjust**. Read-write. BIOS: D18F2x[1,0]9C_x0D0F_0[F,7:0]1F[RxVioLvl]. This field specifies the adjustment signals for the comparator differential amplifier. Setting this field in DCT0 adjusts the comparator for DCT0 and DCT1. Setting this field in DCT1 has no effect. |
| 1:0 | Reserved. |

### D18F2x[1,0]9C_x0D0F_[C,8][1:0]02 Transmit PreDriver Calibration

Cold reset: xxxx_xxxxh. BIOS: See 2.9.3.2.4.

**Table 112: Index address mapping for** D18F2x[1,0]9C_x0D0F_[C,8][1:0]02

| D18F2x[1,0]98[31:0] | Function |
|---|---|

**Table 112: Index address mapping for** D18F2x[1,0]9C_x0D0F_[C,8][1:0]02

| | |
|---|---|
| 0D0F_8002h | Cmd/Addr 0 Pad Group 0 |
| 0D0F_8102h | Cmd/Addr 1 Pad Group 0 |
| 0D0F_C002h | Address Pad Group 0 |

| Bits | Description |
|---|---|
| 31:16 | Reserved. |
| 15 | **ValidTxAndPre: predriver calibration code valid**. Read-write; cleared-when-done. 1=Predriver calibration codes are copied from this register and D18F2x[1,0]9C_x0D0F_[C,8][1:0][12,0E,0A,06] into the associated transmit pad. |
| 14:12 | Reserved. |
| 11:6 | **TxPreP: PMOS predriver calibration code**. Read-write. Specifies the rising edge slew rate of the transmit pad. See: Table 88 [Valid values for D18F2x[1,0]9C_x0D0F_0[F,7:0]02[TxPreP, TxPreN]]. After updating this value, BIOS must program ValidTxAndPre=1 for the change to take effect. |
| 5:0 | **TxPreN: NMOS predriver calibration code**. Read-write. Specifies the falling edge slew rate of the transmit pad. See: Table 88 [Valid values for D18F2x[1,0]9C_x0D0F_0[F,7:0]02[TxPreP, TxPreN]]. After updating this value, BIOS must program ValidTxAndPre=1 for the change to take effect. |

### D18F2x[1,0]9C_x0D0F_[C,8][1:0][12,0E,0A,06] Transmit PreDriver Calibration 2

Cold reset: xxxx_xxxxh. BIOS: See 2.9.3.2.4.

**Table 113: Index address mapping for** D18F2x[1,0]9C_x0D0F_[C,8][1:0][12,0E,0A,06]

| Index | Function | Index | Function |
|---|---|---|---|
| 0D0F_8006h | Cmd/Addr 0 Pad Group 1 | 0D0F_C006h | Address Pad Group 1 |
| 0D0F_800Ah | Cmd/Addr 0 Pad Group 2 | 0D0F_C00Ah | Address Pad Group 2 |
| 0D0F_8106h | Cmd/Addr 1 Pad Group 1 | 0D0F_C00Eh | Address Pad Group 3 |
| 0D0F_810Ah | Cmd/Addr 1 Pad Group 2 | 0D0F_C012h | Address Pad Group 4 |

| Bits | Description |
|---|---|
| 31:12 | Reserved. |
| 11:6 | **TxPreP: PMOS predriver calibration code**. Read-write. Specifies the rising edge slew rate of the transmit pad. See: Table 88 [Valid values for D18F2x[1,0]9C_x0D0F_0[F,7:0]02[TxPreP, TxPreN]]. After updating this value, BIOS must program D18F2x[1,0]9C_x0D0F_[C,8][1:0]02[ValidTxAnd-Pre]=1 for the change to take effect. |
| 5:0 | **TxPreN: NMOS predriver calibration code**. Read-write. Specifies the falling edge slew rate of the transmit pad. See: Table 88 [Valid values for D18F2x[1,0]9C_x0D0F_0[F,7:0]02[TxPreP, TxPreN]]. After updating this value, BIOS must program D18F2x[1,0]9C_x0D0F_[C,8][1:0]02[ValidTxAnd-Pre]=1 for the change to take effect. |

### D18F2x[1,0]9C_x0D0F_812F Addr/Cmd Tri-state Configuration

Cold reset: 0000_00A0h. BIOS: See 2.9.3.9.

| Bits | Description |
|---|---|
| 31:8 | Reserved. |
| 7 | **Add16Tri: MEMADD[16] tri-state**. Read-write. This field specifies tri-state control for the memory address[16] signal. 1=Signal is tri-stated. 0=Signal is not tri-stated. |
| 6 | Reserved. |
| 5 | **Add17Tri: MEMADD[17] tri-state**. Read-write. This field specifies tri-state control for the memory address[17] signal. 1=Signal is tri-stated. 0=Signal is not tri-stated. |
| 4:1 | Reserved. |
| 0 | **PARTri: MEMPAR tri-state**. Read-write. This field specifies tri-state control for the memory parity signal. 1=Signal is tri-stated. 0=Signal is not tri-stated. |

#### D18F2x[1,0]9C_x0D0F_C000 CKE 2.0X Pad Configuration

Cold reset: 0000_0003h. BIOS: See 2.9.3.9.

| Bits | Description |
|---|---|
| 31:9 | Reserved. |
| 8 | **LowPowerDrvStrengthEn: low power drive strength enable**. Read-write. 1=CKE driver is forced to low drive strength when phy is in self-refresh mode. 0=CKE is driven with strength specified by D18F2x[1,0]9C_x0000_0000[CkeDrvStren] when phy is in self-refresh mode. |
| 7:0 | Reserved. |

#### D18F2x[1,0]9C_x0D0F_E003 Phy Calibration Configuration

Cold reset: 0000_0210h.

| Bits | Description |
|---|---|
| 31:15 | Reserved. |
| 14 | **DisAutoComp: disable automatic compensation**. Read-write. BIOS: See 2.9.3 and 2.9.3.2.4. 1=Disable the compensation control state machine. 0=The phy automatic compensation engine is enabled. Setting this bit in DCT0 or DCT1 disables the compensation engine for DCT0 and DCT1. |
| 13 | **DisablePredriverCal: disable predriver calibration**. Read-write. BIOS: See 2.9.3 and 2.9.3.2.4. 1=Disables hardware update of predriver calibration codes. Setting this bit in DCT0 disables the update of predriver calibration codes for DCT0 and DCT1. Setting this bit in DCT1 has no effect. |
| 12:0 | Reserved. |

#### D18F2x[1,0]9C_x0D0F_E006 Phy PLL Lock Time

Cold reset: 0000_0190h.

| Bits | Description |
|---|---|
| 31:16 | Reserved. |
| 15:0 | **PllLockTime: Pll lock time**. Read-write. BIOS: See 2.9.3.2.2. This field specifies the number of 5 ns periods the phy waits for PLLs to lock during a frequency change. |

### D18F2x[1,0]9C_x0D0F_E00A Phy Dynamic Power Mode

Cold reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:15 | Reserved. |
| 14 | **SelCsrPllPdMode: mode**. Read-write. BIOS: 1. 1=CsrPhySrPllPdMode selects power down mode. 0=Reserved. This bit must be programmed the same on both DCTs. |
| 13:12 | **CsrPhySrPllPdMode: CSR phy self refresh power down mode**. Read-write. BIOS: 10b. Selects the PLL power down mode during phy self refresh when SelCsrPllPdMode=1. These bits must be programmed the same on both DCTs.<br>Bits Definition<br>00b No PLL power down<br>01b PLL VCO power down<br>10b PLL regulator power down<br>11b Reserved |
| 11:5 | Reserved. |
| 4 | **SkewMemClk: skew MEMCLK**. Read-write. BIOS: 0. 1=Skew MEMCLK signals with respect to other channel. SkewMemClk must be 0 if D18F2x[1,0]A8[DbeGskMemClkAlignMode]=10b or if D18F2x[1,0]94[DisDramInterface]=1 for either channel. This bit must be set prior to setting D18F2x[1,0]94[MemClkFreqVal] during DRAM initialization. See 2.9.3.2.2 [DRAM Channel Frequency Change]. |
| 3:0 | Reserved. |

### D18F2x[1,0]9C_x0D0F_E013 Phy PLL Regulator Wait Time

Cold reset: 0000_00D8h.

| Bits | Description |
|---|---|
| 31:16 | Reserved. |
| 15:0 | **PllRegWaitTime: PLL regulator wait time**. Read-write. BIOS: See 2.9.3.2. This field specifies the number of 5 ns periods the phy waits for the PLL to become stable when coming out of PLL regulator off power down mode. |

### D18F2x[1,0]A0 DRAM Controller Miscellaneous Register

Reset: 0000_0000h. See 2.9.1 [DCT Configuration Registers] for general programming information about DCT configuration registers.

| Bits | Description |
|---|---|
| 31:0 | Reserved. |

### D18F2xA4 DRAM Controller Temperature Throttle Register

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:3 | Reserved. |

| | |
|---|---|
| 2:1 | **ThrottleEn: DRAM throttle enable**. Read-write. BIOS: See 2.5.6.2 and 2.9.3.5.This field specifies the average utilization of the channel if the EVENT_L pin is asserted. Throttling is accomplished by reducing command issue bandwidth based on historical command throttle values (Precharge, Activate, and AutoPrecharge = 1, Read and Write = 4).<br>Bits Definition<br>00b 100% (No throttling)<br>01b 50% (New commands not issued more frequently than 1 every 2 * value NCLKs.)<br>10b 25% (New commands not issued more frequently than 1 every 4 * value NCLKs.)<br>11b 12.5% (New commands not issued more frequently than 1 every 8 * value NCLKs.) |
| 0 | **DoubleTrefRateEn: double Tref rate enable**. Read-write. 1=Tref forced to 3.9us auto-refresh interval when the EVENT_L pin is asserted. See 2.5.6.2 [EVENT_L]. |

### D18F2x[1,0]A8 DRAM Controller Miscellaneous Register 2

See 2.9.1 [DCT Configuration Registers] for general programming information about DCT configuration registers.

| Bits | Description |
|---|---|
| 31:23 | Reserved. |
| 22:21 | **DbeGskMemClkAlignMode: DBE gasket memclk align mode**. Read-write. Cold reset: 0. BIOS: See 2.9.3 and 2.9.3.2. Specifies the method used to align DDR commands with MEMCLK.<br>Bits Definition<br>00b Command shift.<br>01b Pointer shift.<br>10b Phy MEMCLK shift.<br>11b Reserved |
| 20 | **BankSwap: swap bank address**. Read-write. Reset: 0. BIOS: See 2.9.3.5. 1=Swap the DRAM bank address bits. If (D18F2x110[DctSelIntLvAddr]==100b) then normalized address bits 10:8 are swapped with bits 15:13 else normalized address bits 11:9 are swapped with bits 15:13. This swap happens before D18F2x[1,0]94[BankSwizzleMode] is applied. |
| 19:0 | Reserved. |

### D18F2xAC DRAM Controller Temperature Status Register

Cold reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:1 | Reserved. |
| 0 | **MemTempHot: Memory temperature hot**. Read; Write-1-to-clear. 1=1=One or both of the M[B,A]_EVENT_L pins was asserted indicating the memory temperature exceeded the normal operating limit; the DCT may be throttling the interface to aid in cooling (see D18F2xA4). See 2.5.6.2 [EVENT_L]. |

### D18F2x[1,0]F0 DRAM Controller Extra Data Offset Register

Reset: 8000_0000h.

The DCTs each include an array of registers called D18F2x[1, 0]F4_x[FFF:0], which are defined following D18F2x[1,0]F4. [The DRAM Controller Extra Data Offset Register] D18F2x[1,0]F0 and [The DRAM Con-

troller Extra Data Port] D18F2x[1,0]F4 are used to access D18F2x[1, 0]F4_x[FFF:0]. The register number (i.e., the number that follows "_x" in the register mnemonic) is specified by D18F2x[1,0]F0[DctOffset]. Access to these registers is accomplished as follows:

- Reads:
  - Write the register number to D18F2x[1,0]F0[DctOffset] with D18F2x[1,0]F0[DctAccessWrite]=0.
  - Read the register contents from D18F2x[1,0]F4.

- Writes:
  - Write all 32 bits to the register data to D18F2x[1,0]F4 (individual byte writes are not supported).
  - Write the register number to D18F2x[1,0]F0[DctOffset] with D18F2x[1,0]F0[DctAccessWrite]=1.

| Bits | Description |
|---|---|
| 31 | **DctAccessDone: DRAM controller access done**. Read-only. 1=The access to one of the F2x[1, 0]F4_x[FFFFFFF:0] registers is complete. 0=The access is still in progress. |
| 30 | **DctAccessWrite: DRAM controller read/write select**. Read-write. 0=Read one of the F2x[1, 0]F4_x[FFFFFFF:0] registers. 1=Write one of the F2x[1, 0]F4_x[FFFFFFF:0] registers. |
| 29:28 | Reserved. |
| 27:0 | **DctOffset: DRAM controller offset**. Read-write. |

### D18F2x[1,0]F4 DRAM Controller Extra Data Port

See D18F2x[1,0]F0.

### D18F2x[1,0]F4_x06 DCT Read Timing

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:12 | Reserved. |
| 11:8 | **TwrrdSD: write-to-read timing same DIMM**. Read-write. BIOS: See 2.9.3.4.3 [Twrrd and TwrrdSD (Write-to-Read DIMM Termination Turn-around)]. This specifies the minimum number of cycles from the last clock of *virtual CAS* of the first write operation to the clock in which CAS is asserted for a following read operation to a different chip select on the same DIMM.<br>Bits Definition<br>Ah-0h <TwrrdSD+1> clock(s)<br>Fh-Bh Reserved |
| 7 | **TrdrdScEn: read to read timing same chip select enable**. Read-write. BIOS: See 2.9.3.5. 0=Two reads to the same chip select are issued optimally with a minimum of 0 wait states. 1=Two reads to same chip select are issued with commands separated as specified by the TrdrdSD field. |
| 6:4 | Reserved. |
| 3:0 | **TrdrdSD: read to read timing same DIMM**. Read-write. BIOS: See 2.9.3.4.1 [Trdrd and TrdrdSD (Read-to-Read Timing)]. This field specifies the minimum number of cycles from the last clock of virtual CAS of a first read-burst operation to the clock in which CAS is asserted for a following read-burst operation that is to a different chip select on the same DIMM.<br>Bits Definition<br>8h-0h <TrdrdSD+2> clocks<br>Fh-9h Reserved |

### D18F2x[1,0]F4_x16 DCT Write Timing

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:4 | Reserved. |
| 3:0 | **TwrwrSD: write to write timing same DIMM**. Read-write. BIOS: See 2.9.3.4.2 [Twrwr and TwrwrSD (Write-to-Write Timing)]. This field specifies the minimum number of cycles from the last clock of virtual CAS of the first write-burst operation to the clock in which CAS is asserted for a following write-burst operation that is to a different chip select on the same DIMM.<br>Bits Definition<br>9h-0h <TwrwrSD+1> clock(s)<br>Fh-Ah Reserved |

### D18F2x[1,0]F4_x30 DCT Skip Numerator

Cold reset: 0000_0400h.

| Bits | Description |
|---|---|
| 31:13 | Reserved. |
| 12:0 | **DbeGskFifoNumerator: FIFO skip numerator**. Read-write. BIOS: See 2.9.3.2.2 and 2.9.3.4.7. This field specifies the skip numerator input term to logic which synchronizes between NCLK and MEMCLK. This field must be re-programmed if NCLK or MEMCLK frequency change. |

### D18F2x[1,0]F4_x31 DCT Skip Denominator

Cold reset: 0000_0200h.

| Bits | Description |
|---|---|
| 31:13 | Reserved. |
| 12:0 | **DbeGskFifoDenominator: FIFO skip denominator**. Read-write. BIOS: See 2.9.3.2.2 and 2.9.3.4.7. This field specifies the skip denominator input term to logic which synchronizes between NCLK and MEMCLK. This field must be re-programmed if NCLK or MEMCLK frequency change. |

### D18F2x[1,0]F4_x32 DCT Transmit Fifo Control

Reset: 0000_0404h.

| Bits | Description |
|---|---|
| 31:16 | Reserved. |
| 15 | **DataTxFifoSchedDlyNegSlot1: slot1 data transmit FIFO schedule delay negative**. See: DataTxFifoSchedDlyNegSlot0. |
| 14:13 | Reserved. |
| 12:8 | **DataTxFifoSchedDlySlot1: slot1 data transmit FIFO schedule delay**. Read-write. See: DataTxFifoSchedDlySlot0. |

| | |
|---|---|
| 7 | **DataTxFifoSchedDlyNegSlot0: slot0 data transmit FIFO schedule delay negative**. Read-write. BIOS: See 2.9.3.5 and 2.9.3.4.7. Specifies the direction of delay as specified by DataTxFifoSchedDlySlot0. 1=DCT will delay the pull of write data versus sending of write CAS. 0=DCT will delay write CAS versus pull of write data. This field must be re-programmed if NCLK or MEMCLK frequency change. |
| 6:5 | Reserved. |
| 4:0 | **DataTxFifoSchedDlySlot0: slot0 data transmit FIFO schedule delay**. Read-write. BIOS: See 2.9.3.5 and 2.9.3.4.7. Specifies FIFO slot0 timing for pulling DRAM write data to send to the phy versus sending write CAS to the phy in order to avoid FIFO overflow conditions. If DataTxFifoSchedDlyNegSlot0=1, this field specifies NCLK cycles, else this field specifies MEMCLK cycles. This field must be re-programmed if NCLK or MEMCLK frequency changes.<br>Bits Definition<br>1Fh-00h <DataTxFifoSchedDlySlot0> clock cycle(s). |

**D18F2x[1,0]F4_x40 DRAM Timing 0**

Reset: 0000_0000h. BIOS: See 2.9.3.3.

| Bits | Description |
|---|---|
| 31:30 | Reserved. |
| 29:24 | **Trc: row cycle time**. Read-write. Specifies the minimum time in memory clock cycles from an activate command to another activate command or an auto-refresh command, all to the same chip select bank.<br>Bits Definition<br>03h-00h Reserved.<br>26h-04h <Trc+16> clocks<br>3Fh-27h Reserved |
| 23:21 | Reserved. |
| 20:16 | **Tras: row active strobe**. Read-write. Specifies the minimum time in memory clock cycles from an activate command to a precharge command, both to the same chip select bank.<br>Bits Definition<br>15h-00h <Tras+15> clocks<br>1Fh-16h Reserved |
| 15:12 | Reserved. |
| 11:8 | **Trp: row precharge time**. Read-write. Specifies the minimum time in memory clock cycles from a precharge command to an activate command or auto-refresh command, both to the same bank.<br>Bits Definition<br>9h-0h <Trp+5> clocks<br>Fh-Ah Reserved |
| 7:4 | Reserved. |
| 3:0 | **Trcd: RAS to CAS delay**. Read-write. Specifies the time in memory clock cycles from an activate command to a read/write command, both to the same bank.<br>Bits Definition<br>9h-0h <Trcd+5> clocks<br>Fh-Ah Reserved |

### D18F2x[1,0]F4_x41 DRAM Timing 1

Reset: 0000_0000h. BIOS: See 2.9.3.3.

| Bits | Description |
|---|---|
| 31:19 | Reserved. |
| 18:16 | **Twtr: internal DRAM write to read command delay**. Read-write. Specifies the minimum number of memory clock cycles from a write operation to a read operation, both to the same chip select. This is measured from the rising clock edge following the last non-masked data strobe of the write to the rising clock edge of the next read command.<br>Bits: Definition<br>100b-000b: <Twtr+4> clocks<br>111b-101b: Reserved |
| 15:11 | Reserved. |
| 10:8 | **Trrd: row to row delay (or RAS to RAS delay)**. Read-write. Specifies the minimum time in memory clock cycles between activate commands to different chip select banks.<br>Bits: Definition<br>100b-000b: <Trrd+4> clocks<br>111b-101b: Reserved |
| 7:3 | Reserved. |
| 2:0 | **Trtp: read CAS to precharge time**. Read-write. Specifies the earliest time in memory clock cycles a page can be closed after having been read. Satisfying this parameter ensures read data is not lost due to a premature precharge.<br>Bits: Definition<br>100b-000b: <Trtp+4> clocks for burst length of 32 or 64 bytes<br>111b-101b: Reserved |

### D18F2x[1,0]F4_x83 DCT ODT Control

Reset: 0000_0000h. See 2.9.3.4.5 [DRAM ODT Control].

| Bits | Description |
|---|---|
| 31:15 | Reserved. |
| 14:12 | **WrOdtOnDuration: write ODT on duration**. Read-write. BIOS: 110b. Specifies the number of memory clock cycles that DIMM ODT is asserted for writes.<br>Bits: Definition<br>000b: 0 clocks (Don’t assert ODT)<br>101b-001b: Reserved<br>111b-110b: <WrOdtOnDuration> clocks |
| 11:9 | Reserved. |
| 8 | **WrOdtTrnOnDly: Write ODT Turn On Delay**. Read-write. BIOS: 0b. Specifies the number of memory clock cycles that DIMM ODT assertion is delayed relative to a write CAS.<br>Bits: Definition<br>0b: 0 clocks (ODT asserted with CAS)<br>1b: 1 clocks |
| 7 | Reserved. |

| | |
|---|---|
| 6:4 | **RdOdtOnDuration: Read ODT On Duration**. Read-write. BIOS: 110b. Specifies the number of memory clock cycles that DIMM ODT is asserted for reads.<br>Bits: Definition<br>000b: 0 clocks (Don't assert ODT)<br>101b-001b: Reserved<br>111b-110b: <RdOdtOnDuration> clocks |
| 3 | Reserved. |
| 2:0 | **RdOdtTrnOnDly: Read ODT Turn On Delay**. Read-write. BIOS: MAX(0, D18F2x[1,0]88[Tcl] - D18F2x[1,0]84[Tcwl]). Specifies the number of memory clock cycles that DIMM ODT assertion is delayed relative to read CAS.<br>Bits: Description<br>100b-000b: <RdOdtTrnOnDly> clocks<br>111b-101b: Reserved |

### D18F2x[1,0]F4_x180 DCT ODT Control

Reset: 0000_0000h. BIOS: See Table 22.

| Bits | Description |
|---|---|
| 31:28 | Reserved. |
| 27:24 | **RdOdtPatCs3: read ODT pattern chip select 3**. Read-write. This field represents the state of ODT[3:0] pins when a read occurs to the specified chip select. |
| 23:20 | Reserved. |
| 19:16 | **RdOdtPatCs2: read ODT pattern chip select 2**. Read-write. This field represents the state of ODT[3:0] pins when a read occurs to the specified chip select. |
| 15:12 | Reserved. |
| 11:8 | **RdOdtPatCs1: read ODT pattern chip select 1**. Read-write. This field represents the state of ODT[3:0] pins when a read occurs to the specified chip select. |
| 7:4 | Reserved. |
| 3:0 | **RdOdtPatCs0: read ODT pattern chip select 0**. Read-write. This field represents the state of ODT[3:0] pins when a read occurs to the specified chip select. |

### D18F2x[1,0]F4_x182 DCT ODT Control

Reset: 0000_0000h. BIOS: See Table 22.

| Bits | Description |
|---|---|
| 31:28 | Reserved. |
| 27:24 | **WrOdtPatCs3: write ODT pattern chip select 3**. Read-write. This field represents the state of ODT[3:0] pins when a write occurs to the specified chip select. |
| 23:20 | Reserved. |
| 19:16 | **WrOdtPatCs2: write ODT pattern chip select 2**. Read-write. This field represents the state of ODT[3:0] pins when a write occurs to the specified chip select. |
| 15:12 | Reserved. |
| 11:8 | **WrOdtPatCs1: write ODT pattern chip select 1**. Read-write. This field represents the state of ODT[3:0] pins when a write occurs to the specified chip select. |

| | |
|---|---|
| 7:4 | Reserved. |
| 3:0 | **WrOdtPatCs0: write ODT pattern chip select 0**. Read-write. This field represents the state of ODT[3:0] pins when a write occurs to the specified chip select. |

### D18F2x[1,0]F4_x200 DCT Power Management

Reset: 0000_0002h.

**Table 114: BIOS Recommendations for** D18F2x[1,0]F4_x200**[Txp]**

| Condition | D18F2x[1,0]F4_x200 |
|---|---|
| DdrRate | Txp |
| 800 | 3h |
| 1066, 1333 | 4h |
| 1600 | 5h |
| 1866 | 6h |

**Table 115: BIOS Recommendations for** D18F2x[1,0]F4_x200**[Txpdll]**

| Condition | D18F2x[1,0]F4_x200 |
|---|---|
| DdrRate | Txpdll |
| 667, 800 | 0h |
| 1066 | 3h |
| 1333 | 6h |
| 1600 | Ah |
| 1866 | Dh |

| Bits | Description |
|---|---|
| 31:13 | Reserved. |
| 12:8 | **Txpdll: exit precharge and DLL PD to command delay**. Read-write. BIOS: Table 115. Specifies the minimum time that the DCT waits to issue a command after exiting precharge powerdown mode if the DLL was also disabled.<br>Bits — Definition<br>15h-00h — <Txpdll+10> clocks<br>1Fh-16h — Reserved |
| 7:4 | Reserved. |
| 3:0 | **Txp: exit precharge PD to command delay**. Read-write. BIOS: Table 114. Specifies the minimum time that the DCT waits to issue a command after exiting precharge powerdown mode.<br>Bits — Definition<br>1h-0h — Reserved<br>7h-2h — <Txp> clocks<br>Fh-8h — Reserved |

### D18F2x10C Interleaved Region Base/Limit Register

Reset: 0000_0000h. Enables swapping a region below 4G with the same sized region located at the bottom of

memory.

- The size of the swapped high region must be a integer multiple of 128M, defined to be {D18F2x10C[IntLvRegionBase],000b,000000h} to {D18F2x10C[IntLvRegionLimit],111b,FFFFFFh}.
- The size and location of the low region is defined to be 0000_0000h to {D18F2x10C[IntLvRegionLimit]-D18F2x10C[IntLvRegionBase],111b,FFFFFFh}.
- The swapped high region can not overlap above MSRC001_001A[TOM].
- The swapped region must be all DRAM. I.e. No IO hole.
- Interleaving must be enabled and the DCTs must be of unequal size.
- See D18F2x110[DctSelIntLvEn]. See 2.9.5 [Memory Hoisting] for programming information.

| Bits | Description |
|---|---|
| 31:16 | Reserved. |
| 15:11 | **IntLvRegionLimit: interleaved region limit address**. IF (D18F2x118[C6DramLock] == 1) THEN Read-only. ELSE Read-write. ENDIF. Interleave region limit address [31:27]. |
| 10:8 | Reserved. |
| 7:3 | **IntLvRegionBase: interleaved region base address**. IF (D18F2x118[C6DramLock] == 1) THEN Read-only. ELSE Read-write. ENDIF. Interleave region base address [31:27]. |
| 2:1 | Reserved. |
| 0 | **IntLvRegionEn: interleaved region remap enable**. IF (D18F2x118[C6DramLock] == 1) THEN Read-only. ELSE Read-write. ENDIF. BIOS: Table 30. 1=Enable remapping into the DCT-interleaved region. |

### D18F2x110 DRAM Controller Select Low Register

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:24 | Reserved. |
| 23:11 | **DctSelBaseAddr[39:27]: DRAM controller select base address bits[39:27]**. IF (D18F2x118[C6DramLock] == 1) THEN Read-only. ELSE Read-write. ENDIF. This field delineates the address range of the two DCTs by specifying the base address of the upper address range. See 2.9.5 [Memory Hoisting] for additional programming information. |
| 10 | **MemCleared: memory cleared**. Read-only. 1=Memory has been cleared since the last warm reset. This bit is set by MemClrInit. See MemClrInit below. |
| 9 | **MemClrBusy: memory clear busy**. Read-only. 1=Memory clear operation in either of the DCTs is in progress. Reads or writes to DRAM while the memory clear operation is in progress result in undefined behavior. |
| 8 | **DramEnable: DRAM enabled**. Read-only. 1=All of the used DCTs are initialized (see 2.9.3.6 [DRAM Device Initialization]) or have exited from self refresh (D18F2x[1,0]90[ExitSelfRef] transitions from 1 to 0). |

<table>
<tr><td>7:6</td><td>DctSelIntLvAddr[1:0]: DRAM controller select channel interleave address bit. IF (D18F2x118[C6DramLock] == 1) THEN Read-only. ELSE Read-write. ENDIF. BIOS: 101b. This specifies how interleaving is selected between the DCTs. In all cases, if the select function is low, then DCT0 is selected; if the select function is high, then DCT1 is selected. DctSelIntLvAddr[2:0] = {D18F2x114[DctSelIntLvAddr[2], D18F2x110[DctSelIntLvAddr[1:0]]}. The select functions are:<br><br>
<table>
<tr><th>DctSelIntLvAddr</th><th>DCT Select Function</th></tr>
<tr><td>000b</td><td>Address bit 6.</td></tr>
<tr><td>001b</td><td>Address bit 12.</td></tr>
<tr><td>010b</td><td>Hash: exclusive OR of address bits[20:16, 6].</td></tr>
<tr><td>011b</td><td>Reserved.</td></tr>
<tr><td>100b</td><td>Address bit 8.</td></tr>
<tr><td>101b</td><td>Address bit 9.</td></tr>
<tr><td>110b</td><td>Reserved.</td></tr>
<tr><td>111b</td><td>Reserved.</td></tr>
</table>
If the internal GPU is enabled only encodings 001b, 100b, and 101b are supported.<br>
If (D18F2x[1,0]A8[BankSwap]==1 and DctSelIntLvEn==0) then this field must not be set to 100b.</td></tr>
<tr><td>5:4</td><td>Reserved.</td></tr>
<tr><td>3</td><td>MemClrInit: memory clear initialization. IF (D18F2x118[C6DramLock] == 1) THEN Read-only. ELSE Read-write; cleared-by-hardware. ENDIF. 1=The processor writes 0’s to all locations of system memory attached to the processor as follows and sets the MemCleared bit:<br>
• If D18F1xF0[DramHoleValid]=0 then memory is cleared:<br>
  • from 0 ({D18F1x40[DramBase[39:24]], 00_0000h}) to {D18F1x40[DramLimit[39:24]], FF_FFFFh}.<br>
• If D18F1xF0[DramHoleValid]=1 then memory is cleared:<br>
  • from 0 ({D18F1x40[DramBase[39:24]], 00_0000h}) to ({00h,D18F1xF0[DramHole-Base[31:24]],000000h}-1).<br>
  • from 01_00000000h (4 GB) to {D18F1x40[DramLimit[39:24]], FF_FFFFh}.<br>
  • Undefined behavior may result if the DRAM hole is enabled (D18F1xF0[DramHoleValid]=1) and D18F1x40[DramLimit[39:32]]=00h. (The DRAM hole is enabled and the DRAM limit is below 4 GB.)<br>
• The status of the memory clear operation can be determined by reading the MemClrBusy and Mem-Cleared bits. This command is ignored if MemClrBusy=1 when the command is received.<br>
• BIOS must set the following registers before setting MemClrInit:<br>
  • [The DRAM Base Register] D18F1x40<br>
  • [The DRAM Hole Address Register] D18F1xF0<br>
  • [The DRAM Bank Address Mapping Register] D18F2x[1,0]80<br>
  • [The DRAM CS Base Address Registers] D18F2x[1,0][4C:40]<br>
  • [The DRAM CS Mask Register] D18F2x[1,0][64:60]<br>
  • [The DRAM Controller Select Low Register] D18F2x110<br>
  • [The DRAM Controller Select High Register] D18F2x114<br>
DramEnable must be set before setting MemClrInit. The memory prefetcher (see D18F2x11C) must be disabled before memory clear initialization and then can be re-enabled when MemCleared=1.</td></tr>
</table>

| | |
|---|---|
| 2 | **DctSelIntLvEn: DRAM controller interleave enable**. IF (D18F2x118[C6DramLock] == 1) THEN Read-only. ELSE Read-write. ENDIF. BIOS: Table 30 and 2.9.3.5. 1=Channel interleave is enabled; D18F2x114[DctSelIntLvAddr] specifies which address bit is used to select between DCT0 and DCT1; this applies from the base system memory address (specified by [The DRAM Base Register] D18F1x40) to DctSelBaseAddr (if enabled). If the amount of memory connected to each of the DCTs is different, then channel interleaving may be supported across the address range that includes both DCTs, the top of which is specified by DctSelBaseAddr; the remainder of the address space, above DctSelBaseAddr, would then be allocated to only the DCT connected to the larger amount of memory, specified by DctSelHi. The interleaved region can be moved from its default location using D18F2x10C. |
| 1 | **DctSelHi: DRAM controller high select**. IF (D18F2x118[C6DramLock] == 1) THEN Read-only. ELSE Read-write. ENDIF. If DctSelHiRngEn is set, this specifies which DCT receives accesses with addresses in the high range (greater than or equal to DctSelBaseAddr). The high range DCT is called DctHi, the low range DCT is called DctLo. 0=High addresses go to DCT0. 1=High addresses go to DCT1. |
| 0 | **DctSelHiRngEn: DRAM controller select high range enable**. IF (D18F2x118[C6DramLock] == 1) THEN Read-only. ELSE Read-write. ENDIF. 1=Enables addresses greater than or equal to DctSelBaseAddr[39:27] to be used to select between DCT0 and DCT1; DctSelHi specifies which DCT occupies the high range. |

### D18F2x114 DRAM Controller Select High Register

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:24 | Reserved. |
| 23:10 | **DctSelBaseOffset[39:26]: DRAM controller select base offset address bits[39:26]**. IF (D18F2x118[C6DramLock] == 1) THEN Read-only. ELSE Read-write. ENDIF. When memory hosting is enabled, this value is subtracted from the physical address of certain transactions before being passed to the DCT. See 2.9.5 [Memory Hoisting] for programming information. |
| 9 | **DctSelIntLvAddr[2]: DRAM controller select channel interleave address bit**. IF (D18F2x118[C6DramLock] == 1) THEN Read-only. ELSE Read-write. ENDIF. See D18F2x110[DctSelIntLvAddr[1:0]] for detailed description. |
| 8:0 | Reserved. |

### D18F2x118 Memory Controller Configuration Low Register

This register indicates the priority of request types. Variable priority requests enter the northbridge as medium priority and are promoted to high priority if they have not been serviced in the time specified by MctVarPriCntLmt. This feature may be useful for isochronous IO traffic. If isochronous traffic is specified to be high priority, it may have an adverse effect on the bandwidth and performance of the devices associated with the other types of traffic. However, if isochronous traffic is specified as medium priority, the processor may not be able to meet the isochronous bandwidth and latency requirements. The variable priority allows the memory controller to optimize DRAM transactions until isochronous traffic reaches a time threshold and must be serviced more quickly.

If a write requires a read-modify-write, arbitration occurs separately for the read and the write and the read has the same priority level as the write. If the priority of the write is changed for a read-modify-write then the priority of the read is changed as well to maintain the same priority was the write.

<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>31:28</td><td>MctVarPriCntLmt: variable priority time limit. Read-write. Reset: 0h.
<table>
<tr><th>Bits</th><th>Definition</th><th>Bits</th><th>Definition</th></tr>
<tr><td>0000b</td><td>80ns</td><td>1000b</td><td>720ns</td></tr>
<tr><td>0001b</td><td>160ns</td><td>1001b</td><td>800ns</td></tr>
<tr><td>0010b</td><td>240ns</td><td>1010b</td><td>880ns</td></tr>
<tr><td>0011b</td><td>320ns</td><td>1011b</td><td>960ns</td></tr>
<tr><td>0100b</td><td>400ns</td><td>1100b</td><td>1040ns</td></tr>
<tr><td>0101b</td><td>480ns</td><td>1101b</td><td>1120ns</td></tr>
<tr><td>0110b</td><td>560ns</td><td>1110b</td><td>1200ns</td></tr>
<tr><td>0111b</td><td>640ns</td><td>1111b</td><td>1280ns</td></tr>
</table></td></tr>
<tr><td>27:20</td><td>Reserved.</td></tr>
<tr><td>19</td><td>C6DramLock. Write-1-only. Reset: 0. BIOS: See 2.5.3.2.9 and 2.9.6. 1=The following registers are read-only:<br>• D18F1x44<br>• D18F1xF0<br>• D18F2x[1,0][4C:40]<br>• D18F2x[1,0][64:60]<br>• D18F2x[1,0]80<br>• D18F2x10C<br>• D18F2x110<br>• D18F2x114<br>• D18F4x12C</td></tr>
<tr><td>18:12</td><td>Reserved.</td></tr>
<tr><td>11:10</td><td>MctPriWr: default write priority. Read-write. Reset: 01b. BIOS: 01b. See: MctPriCpuRd.</td></tr>
<tr><td>9:8</td><td>MctPriDefault: default non-write priority. Read-write. Reset: 00b. BIOS: 00b. See: MctPriCpuRd.</td></tr>
<tr><td>7:6</td><td>MctPriHiWr: high-priority VC set write priority. Read-write. Reset: 00b. BIOS: 10b. See: MctPriCpuRd.</td></tr>
<tr><td>5:4</td><td>MctPriHiRd: high-priority VC set read priority. Read-write. Reset: 10b. BIOS: 10b. See: MctPriCpuRd.</td></tr>
<tr><td>3:2</td><td>MctPriCpuWr: CPU write priority. Read-write. Reset: 01b. BIOS: 01b. See: MctPriCpuRd.</td></tr>
<tr><td>1:0</td><td>MctPriCpuRd: CPU read priority. Read-write. Reset: 00b. BIOS: 00b.
<table>
<tr><th>Bits</th><th>Definition</th><th>Bits</th><th>Definition</th></tr>
<tr><td>00b</td><td>Low</td><td>10b</td><td>High</td></tr>
<tr><td>01b</td><td>Medium</td><td>11b</td><td>Variable</td></tr>
</table></td></tr>
</table>

### D18F2x11C Memory Controller Configuration High Register

The two main functions of this register are to control write bursting and memory prefetching.

**Write bursting**. DctWrLimit specifies how writes may be burst from the DCT to improve DRAM efficiency. Bursting writes improves DRAM efficiency by minimizing the read-to-write turnaround time and the interference that non-latency critical stores have on latency critical loads. When the number of writes in the DCT reaches the value specified in DctWrLimit, then they become eligible for scheduling. Once eligible for sched-

uling, the priority based reorder algorithm picks the optimal write to increase DRAM bandwidth.

Rules regarding write bursting:
- Write bursting mode only applies to low-priority writes. Medium and high priority writes are not withheld from the DCTs for write bursting.
- If write bursting is enabled, writes stay in the DCT until the threshold specified by DctWrLimit is reached. Once the threshold is reached, all writes in DCT are converted to medium-priority. Low-priority and medium-priority reads are not eligible until all converted medium-priority writes are scheduled.
- Any write in the DCT that matches the address of a subsequent access is promoted to either medium priority or the priority of the subsequent access, whichever is higher.

**Memory prefetching**. The DRAM prefetcher detects stride patterns in the stream of requests and then, for predictable stride patterns, generates prefetch requests. A stride pattern is a pattern of requests through system memory that are the same number of cache lines apart. The prefetcher supports strides of -4 to +4 cache lines, which can include alternating patterns (e.g. +1, +2, +1, +2), and can prefetch two strides ahead depending on the confidence. The prefetcher tracks up to 8 stride patterns simultaneously. Each of these stride patterns has a confidence level associated with it that is modified by how many requests match the stride pattern and is used to determine whether to fetch two strides ahead. The prefetcher behaves according to the following rules for each request:
- A prefetch request may be generated only when a demand request is received. When a demand address matches a tracked address and its stride pattern, then:
  - The confidence level for that stride pattern is incremented if less than PrefConfSat.
  - At this time, if the confidence level < PrefConf, no prefetch request is generated. If the confidence level >= PrefConf, one prefetch request is generated which is two consecutive strides ahead in the pattern.
  - Before the prefetch request is issued to the DCT, the prefetch address is checked that it still falls within the same 4 KB page as the request. If it the prefetch address crosses into a different 4 KB page, then the prefetch is squashed and the stride pattern is deallocated.
- Each time a request is received within +/- 4 cache lines of the last recorded address in the pattern and does not match the current stride pattern, then the confidence level is decreased by one.
- Each request that is within the same 4 KB page but outside the -4 to +4 cache line range (including the exact same cache line) of the last requested cache line of all the stride patterns tracked is ignored.
- Each request that is not within the same 4 KB page as the last requested cache line of the stride patterns tracked initiates a new stride pattern by displacing one of the existing least-recently-used stride patterns.
  - The confidence level is initialized to 0 for all new stride patterns.

| Bits | Description |
|---|---|
| 31 | Reserved. |
| 30 | **FlushWr: flush writes command**. Read; Write-1-only; cleared-when-done. Reset: 0. BIOS: 0. Setting this bit causes write bursting to be cancelled and all outstanding writes to be flushed to DRAM. |
| 29 | **FlushWrOnStpGnt: flush writes on stop-grant**. Read-write. Reset: 0. BIOS: 0. 1=Causes write bursting to be cancelled and all outstanding writes to be flushed to DRAM when in the stop-grant state. This bit should be set to ensure writes are drained to DRAM before reset is asserted for the suspend-to-RAM state. |
| 28:25 | Reserved. |

| | |
|---|---|
| 24:22 | **PrefConf: prefetch two-ahead confidence**. Read-write. Reset: 011b. BIOS: 1h. Confidence level required to issue one prefetch which is 2 strides ahead. Steady state keeps 2 prefetches ahead.<br>Bits: Definition<br>000b: Reserved<br>111b-001b: <PrefConf*2> |
| 21:20 | Reserved. |
| 19:18 | **PrefConfSat: prefetch confidence saturation**. Read-write. Reset: 00. BIOS: 0h. Specifies the point at which prefetch confidence level saturates and stops incrementing.<br>Bits: Definition<br>00b: 15<br>01b: 7<br>10b: 3<br>11b: Reserved |
| 17:15 | Reserved. |
| 14 | **PrefCpuRdSzDis: prefetch CPU sized read disable**. Read-write. Reset: 1. BIOS: 1. 1=Disable generating prefetch requests for sized read requests from the cores and prevents sized read requests from the cores from consuming prefetch data. This bit has no effect if PrefCpuDis=1. |
| 13 | Reserved. |
| 12 | **PrefCpuDis: prefetch CPU access disable**. Read-write. Reset: 1. BIOS: See 2.9.3.5. 1=Disables core requests from triggering prefetch requests and prevents core requests from consuming prefetch data. |
| 11:7 | Reserved. |
| 6:2 | **DctWrLimit: memory controller write-burst limit**. Read-write. Reset: 18h. BIOS: See 2.9.3.5. Specifies the number of low-priority writes held in the memory controller queue before they are burst into the DCTs.<br>Bits: Definition<br>11h-00h: Reserved.<br>1Eh-12h: <32-DctWrLimit><br>1Fh: Write bursting disabled |
| 1:0 | Reserved. |

### D18F2x1C0 DRAM Training Control

Reset: 0000_0000h. See 2.9.3.7.6 [DRAM Training Pattern Generation].

| Bits | Description |
|---|---|
| 31:24 | Reserved. |
| 23 | **RdTrainGo: read training go**. Read-write; cleared-when-done. 1=Initiate the data transfer from the DRAM interface to the read training buffer. |
| 22 | **RdDramTrainMode: DRAM read training mode**. Read-write. 1=Enable read training mode. When RdDramTrainMode=1, WrDramTrainMode must be 0. |
| 21 | **AltAddrEn: alternate address enable**. Read-write. 1=Enable alternative address mode. See 2.9.3.7.6.2 [Alternative Address Mode]. 0=DRAM address specified by {D18F2x1CC[TrainAddrPtr[39:38]], D18F2x1C8[TrainAddrPtr[37:6]]}. |

| | |
|---|---|
| 20 | **DramTrainPdbDis: DRAM training prefetch buffer disable**. Read-write. BIOS: See 2.9.3.5. 1=Disable the use of additional prefetch buffers for read continuous pattern generation. 0=Allow use of additional prefetch buffers. If this bit is 0, all prefetching must be disabled by setting D18F2x11C[PrefCpuRdSzDis, PrefCpuDis]=11b. |
| 19:18 | Reserved. |
| 17:2 | **TrainLength: length in cache lines**. Read-write. Specifies the number of cache lines transferred from the write training buffer to the DRAM interface or from the DRAM interface to the read training buffer. |
| 1 | **WrTrainGo: write training go**. Read-write; cleared-when-done. 1=Initiate the data transfer from the write training buffer to the DRAM interface. |
| 0 | **WrDramTrainMode: DRAM write training mode**. Read-write. 1=Enable write training mode. When WrDramTrainMode=1, RdDramTrainMode must be 0. |

### D18F2x1C8 DRAM Training Address Pointer Low

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:0 | **TrainAddrPtr[37:6]: DRAM training address pointer bits[37:6]**. Read-write. Specifies the lower bits of the DRAM address pointer in DRAM training mode. See D18F2x1CC. |

### D18F2x1CC DRAM Training Address Pointer High

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:26 | **AltAddr3PtrIt: DRAM training alternate address pointer 3 iterations**. Read-write. See AltAddr1PtrIt. Specifies the number of times to iterate using AltAddr3Ptr before switching to TrainAddrPtr. |
| 25:24 | **AltAddr3Ptr[39:38]: DRAM training alternate address pointer 3bits[39:38]**. Read-write. See AltAddr1Ptr[39:38]. |
| 23:18 | **TrainAddrPtrIt: DRAM training address pointer iterations**. Read-write. Specifies the number of times to iterate using TrainAddrPtr before switching to AltAddr1Ptr. The number of cache lines transferred using TrainAddrPtr is TrainAddrPtrIt + 1. See 2.9.3.7.6.2 [Alternative Address Mode]. |
| 17:16 | **TrainAddrPtr[39:38]: DRAM training address pointer bits[39:38]**. Read-write. Specifies the upper bits of the DRAM address pointer in DRAM training mode. See D18F2x1C8.<br>• TrainAddrPtr[39:6]={TrainAddrPtr[39:38], D18F2x1C8[TrainAddrPtr[37:6]]};<br>See 2.9.3.7.6.1 [Continuous Pattern Generation] and 2.9.3.7.6.2 [Alternative Address Mode]. |
| 15:10 | **AltAddr2PtrIt: DRAM training alternate address pointer 2 iterations**. Read-write. See AltAddr1PtrIt. |
| 9:8 | **AltAddr2Ptr[39:38]: DRAM training alternate address pointer 2 bits[39:38]**. Read-write. See AltAddr1Ptr[39:38]. |

| | |
|---|---|
| 7:2 | **AltAddr1PtrIt: DRAM training alternate address pointer 1 iterations**. Read-write. Specifies the number of times to iterate using AltAddr1Ptr before switching to AltAddr2Ptr. The number of cache lines transferred using AltAddr1Ptr is AltAddr1PtrIt + 1. See 2.9.3.7.6.2 [Alternative Address Mode]. |
| 1:0 | **AltAddr1Ptr[39:38]: DRAM training alternate address pointer 1 bits[39:38]**. Read-write. Specifies the upper bits of the DRAM first alternate address pointer. See D18F2x[1E0:1D8].<br>• AltAddr1Ptr[39:6]={AltAddr1Ptr[39:38], D18F2x[1E0:1D8][AltAddr1Ptr[37:6]]};<br>See 2.9.3.7.6.2 [Alternative Address Mode]. |

### D18F2x1D0 DRAM Write Training Buffer Address

Reset: 0000_0000h. Read-write.

| Bits | Description |
|---|---|
| 31:10 | Reserved. |
| 9:0 | **WrTrainBufAddr: write training buffer address**. Read-write. Specifies the training data start address in the 1024 dword write training buffer. It is incremented by hardware after a write to D18F2x1D4. BIOS must program this register prior to filling the buffer or setting D18F2x1C0[WrTrainGo]. BIOS must write the lower four bits to 0h to begin on a cache line boundary. |

### D18F2x1D4 DRAM Write Training Data

Write-only. Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:0 | **WrTrainBufDat: write training buffer data**. Writing to this register writes the next dword of the training pattern to the training write buffer and increments D18F2x1D0[WrTrainBufAddr]. See D18F2x1D0[WrTrainBufAddr]. |

### D18F2x[1E0:1D8] DRAM Training Alternate Address Pointer Low

Reset: 0000_0000h.

**Table 116: Address mapping for D18F2x[1E0:1D8].**

| Register | Function |
|---|---|
| D18F2x1D8 | Alternate Address 1 |
| D18F2x1DC | Alternate Address 2 |
| D18F2x1E0 | Alternate Address 3 |

| Bits | Description |
|---|---|
| 31:0 | **AltAddrPtr[37:6]: DRAM training alternate address pointer bits[37:6]**. Read-write. If D18F2x1C0[AltAddrEn]=1 specifies the lower bits of the DRAM alternate address pointer in DRAM training mode. See D18F2x1CC. |

#### D18F2x1E8 DRAM Training Status

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:16 | Reserved. |
| 15:8 | **TrainCmpSts2: DRAM training compare status 2**. Read-only; updated-by-hardware. Contains the comparison results between write and read training data. The last beat of read data in all cache line transfers is ignored. The read data of beats 0 to 6 are compared against the write data of beats 1 to 7. 1=comparison result is a miscompare. TrainCmpSts2[0] contains result for byte lane 0, TrainCmpSts2[7] contains the result of byte lane 7. Hardware clears the field when D18F2x1C0[RdDramTrainMode] changes from 0 to 1. |
| 7:0 | **TrainCmpSts: DRAM training compare status**. Read-only; updated-by-hardware. Contains the comparison results between write and read training data. 1=comparison result is a miscompare. TrainCmpSts[0] contains result for byte lane 0, TrainCmpSts[7] contains the result of byte lane 7. Hardware clears the field when D18F2x1C0[RdDramTrainMode] changes from 0 to 1. |

### 3.10 Device 18h Function 3 Configuration Registers

See 3.1 [Register Descriptions and Mnemonics] for a description of the register naming convention. See 2.7 [Configuration Space] for details about how to access this space.

#### D18F3x00 Device/Vendor ID Register

Reset: 1703_1022h.

| Bits | Description |
|---|---|
| 31:16 | **DeviceID: device ID**. Read-only. |
| 15:0 | **VendorID: vendor ID**. Read-only. |

#### D18F3x04 Status/Command Register

IF (D18F3xE8[SvmCapable]==1) THEN Reset: 0010_0000h. ELSE Reset: 0000_0000h. ENDIF.

| Bits | Description |
|---|---|
| 31:16 | **Status**. Read-only. |
| 15:0 | **Command**. Read-only. |

#### D18F3x08 Class Code/Revision ID Register

Reset: 0600_0000h.

| Bits | Description |
|---|---|
| 31:8 | **ClassCode**. Read-only. Provides the host bridge class code as defined in the PCI specification. |
| 7:0 | **RevID: revision ID**. Read-only. |

### D18F3x0C Header Type Register

Reset: 0080_0000h.

| Bits | Description |
|---|---|
| 31:0 | **HeaderTypeReg**. Read-only. These bits are fixed at their default values. The header type field indicates that there multiple functions present in this device. |

### D18F3x34 Capability Pointer Register

| Bits | Description |
|---|---|
| 31:8 | Reserved. |
| 7:0 | **CapPtr**. Read-only. IF (D18F3xE8[SvmCapable]==1) THEN Reset: F0h. ELSE Reset: 00h. ENDIF. Specifies the configuration-space offset of the capabilities pointer. |

### D18F3x40 MCA NB Control Register

Reset: 0000_0000h.

The machine check registers are used to configure the Machine Check Architecture (MCA) functions of the NB hardware and to provide a method for the NB to report errors in a way compatible with MCA. All of the NB MCA registers, except [The MCA NB Configuration Register] D18F3x44, are accessible through the MCA-defined MSR method, as well as through PCI configuration space.

[The MCA NB Control Register] D18F3x40 enables MCA reporting of each error checked by the NB. The global MCA error enables must also be set through [The Global Machine Check Exception Reporting Control Register (MCG_CTL)] MSR0000_017B. The error enables in this register only affect error reporting through MCA. Actions which the NB may take in addition to MCA reporting are enabled through [The MCA NB Configuration Register] D18F3x44.

Correctable and uncorrectable errors are logged in [The MCA NB Status Low Register] D18F3x48, [The MCA NB Status High Register] D18F3x4C, and [The MCA NB Address Low Register] D18F3x50 as they occur, as specified by D18F3x4C[Over]. Uncorrectable errors immediately result in a Machine Check exception.

| Bits | Description |
|---|---|
| 31:26 | Reserved. |
| 25 | **McaUsPwDatErrEn: MCA upstream data error enable**. Read-write. 1=Enables MCA reporting of upstream posted writes in which the EP bit is set. |
| 24:18 | Reserved. |
| 17 | **DataEn: data error reporting enable**. Read-write. 1=Enables MCA reporting of completion packets with the EP bit set . |
| 16 | **ProtEn: protocol error reporting enable**. Read-write. 1=Enables MCA reporting of internal protocol errors. |
| 15:14 | Reserved. |
| 13 | **DevErrEn: DEV error reporting enable**. Read-write. 1=Enables MCA reporting of SVM DEV errors. |

| | |
|---|---|
| 12 | **WDTRptEn: watchdog timer error reporting enable**. Read-write. 1=Enables MCA reporting of watchdog timer errors. The watchdog timer checks for NB system accesses for which a response is expected but no response is received. See [The MCA NB Configuration Register] D18F3x44 for information regarding configuration of the watchdog timer duration. This bit does not affect operation of the watchdog timer in terms of its ability to complete an access that would otherwise cause a system hang. This bit only affects whether such errors are reported through MCA. |
| 11 | **AtomicRMWEn: atomic read-modify-write error reporting enable**. Read-write. 1=Enables MCA reporting of atomic read-modify-write (RMW) commands received from a link. Atomic RMW commands are not supported. An atomic RMW command results in a link error response being generated back to the requesting IO device. The generation of the link error response is not affected by this bit. |
| 10 | Reserved. |
| 9 | **TgtAbortEn: target abort error reporting enable**. Read-write. 1=Enables MCA reporting of target aborts to a link. The NB returns an error response back to the requestor with any associated data all 1s independent of the state of this bit. |
| 8 | **MstrAbortEn: master abort error reporting enable**. Read-write. 1=Enables MCA reporting of master aborts to a link. The NB returns an error response back to the requestor with any associated data all 1s independent of the state of this bit. |
| 7:6 | Reserved. |
| 5 | **SyncFloodEn: sync flood error reporting enable**. Read-write. 1=Enables MCA reporting of sync flood errors. |
| 4:0 | Reserved. |

**D18F3x44 MCA NB Configuration Register**

| Bits | Description |
|---|---|
| 31 | **NbMcaLogEn: NB MCA log enable**. Read-write. Reset: 0. 1=Enables logging (but not reporting) of NB MCA errors even if MCA is not globally enabled. |
| 30 | Reserved. |
| 29 | **DisMstAbtCpuErrRsp: master abort CPU error response disable**. Read-write. Reset: 0. 1=Disables master abort reporting through the MCA error-reporting banks. Master abort errors do not cause a sync flood when this bit is set. |
| 28 | **DisTgtAbtCpuErrRsp: target abort CPU error response disable**. Read-write. Reset: 0. 1=Disables target abort reporting through the MCA error-reporting banks. Target abort errors do not cause a sync flood when this bit is set. |

| | |
|---|---|
| 27 | **NbMcaToMstCpuEn: machine check errors to master CPU only**. Read-write. Reset: 0. BIOS: 1. 1=MCA errors in CMP device are only reported to core 0, and the NB MCA registers in MSR space (MSR0000_0408, MSR0000_0409, MSR0000_040A, MSR0000_0410, MSR0000_0411 and MSR0000_0412) are only accessible from core 0; reads of these MSRs from other cores return 0 and writes are ignored. This field does not affect PCI-defined configuration space accesses to these registers, which are accessible from all cores. See 3.1 [Register Descriptions and Mnemonics] for a description of MSR space and 3 [Registers] for PCI-defined configuration space.<br>0=MCA errors may be reported to the CPU that originated the request, if applicable and known.<br><br>• When the CPU which originates a request is known, it is stored in D18F3x4C[ErrCPU], regardless of the setting of NbMcaToMstCpuEn.<br>• If IO originated the request, then the error is reported to core 0, regardless of the setting of NbMcaToMstCpuEn. |
| 26 | **CorrMcaExcEn: correctable error MCA exception enable**. Read-write. Reset: 0. 1=Correctable errors that are enabled for checking and logging cause a machine check exception (reporting) in addition to being logged. |
| 25 | **DisPciCfgCpuErrRsp: PCI configuration CPU error response disable**. Read-write. Reset: 0. 1=Disables generation of an error response to the core on detection of a master abort, target abort, or data error condition, and disables logging and reporting through the MCA error-reporting banks for PCI configuration accesses. Also, for NB WDT errors on PCI configuration accesses, this prevents sending an error response to the core, but does not affect logging and reporting of the NB WDT error. See D18F3x180[DisPciCfgCpuMstAbtRsp], which applies only to master aborts. |
| 24 | **IoRdDatErrEn: IO read data error log enable**. Read-write. Reset: 0. 1=Enables logging and reporting of read data errors (link defined master aborts, target aborts, and data error) for data destined for IO devices. 0=Read data errors for transactions from IO devices are not logged by MCA, although error responses may still be generated to the requesting IO device. |
| 23:22 | Reserved. |
| 21 | **SyncOnAnyErrEn: sync flood on any error enable**. Read-write. Reset: 0. 1=Enables generating a sync flood on detection of any NB MCA error that is uncorrectable. |
| 20 | **SyncOnWDTEn: sync flood on watchdog timer error enable**. Read-write. Reset: 0. BIOS: 1. 1=Enables generating a sync flood on detection of a watchdog timer error. |
| 19:14 | Reserved. |
| 13:12 | **WDTBaseSel: watchdog timer time base select**. Read-write. Reset: 0. BIOS: 0. Selects the time base used by the watchdog timer. The counter selected by WDTCntSel determines the maximum count value in the time base selected by WDTBaseSel.<br>Bits / Definition / Bits / Definition<br>00b / 1.31 ms / 10b / 80 ns<br>01b / 1.28 us / 11b / Reserved |

| | |
|---|---|
| 11:9 | **WDTCntSel[2:0]: watchdog timer count select bits[2:0]**. Read-write. Reset: 0. BIOS: 0. Selects the count used by the watchdog timer. WDTCntSel[3:0]={D18F3x180[WDTCntSel[3]], D18F3x44[WDTCntSel[2:0]]}. The counter selected by WDTCntSel determines the maximum count value in the time base selected by WDTBaseSel. WDTCntSel is encoded as:<br>Bits Definition Bits Definition<br>0h 4095 6h 63<br>1h 2047 7h 31<br>2h 1023 8h 8191<br>3h 511 9h 16383<br>4h 255 Fh-Ah Reserved<br>5h 127<br>Because WDTCntSel is split between two registers, care must be taken when programming WDTCnt-Sel to ensure that a reserved value is never used by the watchdog timer or undefined behavior could result. |
| 8 | **WDTDis: watchdog timer disable**. Read-write. Cold reset: 0. 1=Disables the watchdog timer. The watchdog timer is enabled by default and checks for NB system accesses for which a response is expected and where no response is received. If such a condition is detected the outstanding access is completed by generating an error response back to the requestor. An MCA error may also be gener-ated if enabled in [The MCA NB Control Register] D18F3x40. |
| 7 | **IoErrDis: IO error response disable**. Read-write. Reset: 0. 1=Disables setting either Error bit in link response packets to IO devices on detection of a target abort, master abort, or data error condition. |
| 6 | **CpuErrDis: CPU error response disable**. Read-write. Reset: 0. 1=Disables generation of a read data error response to the core on detection of a target abort, master abort or data error condition. |
| 5 | **IoMstAbortDis: IO master abort error response disable**. Read-write. Reset: 0. 1=Signals target abort instead of master abort in link response packets to IO devices on detection of a master abort error condition. When IoMstAbortDis and D18F3x180[MstAbtChgToNoErrs] are both set, D18F3x180[MstAbtChgToNoErrs] takes precedence. |
| 4:3 | Reserved. |
| 2 | Reserved. |
| 1 | **CpuRdDatErrEn: CPU read data error log enable**. Read-write. Reset: 0. 1=Enables logging and reporting of read data errors (master aborts, target aborts, and data error) for data destined for the CPU. This bit should be clear if read data error logging is enabled for the remaining error reporting blocks in the CPU. Logging the same error in more than one block may cause a single error event to be treated as a multiple error event and cause the CPU to enter shutdown. |
| 0 | Reserved. |

### D18F3x48 MCA NB Status Low Register

Cold reset: xxxx_xxxxh.

| Bits | Description |
|---|---|
| 31:21 | Reserved. |
| 20:16 | **ErrorCodeExt: extended error code**. Read-write; updated-by-hardware. Logs the extended error code when an error is detected. See Table 118 for the ErrorCodeExt encodings. |
| 15:0 | **ErrorCode: error code**. Read-write; updated-by-hardware. Logs an error code when an error is detected. See Table 44, Table 45, Table 46, Table 47, Table 48 for the ErrorCode encodings. |

The NB is capable of reporting the following errors:

**Table 117: NB error descriptions**

| Error Type | Description | Control Bits (D18F3x40) |
|---|---|---|
| Sync Flood | Unrecoverable error condition. | SyncFloodEn |
| Mst Abort | Master abort seen as result of link operation. Reasons for this error include requests to non-existent addresses, and requesting extended addresses while extended mode disabled (see D18F0x68[CHtExtNodeEn]). The NB returns an error response back to the requestor with any associated data all 1s independent of the state of the control bit. | MstrAbortEn |
| Target Abort | Target abort seen as result of link operation. The NB returns an error response back to the requestor with any associated data all 1s independent of the state of the control bit. | TgtAbortEn |
| WDT Error | NB WDT timeout due to lack of progress. The NB WDT monitors transaction completions. A transaction that exceeds the programmed time limit reports errors via the MCA. The cause of error may be another device which failed to respond. | WDTRptEn |
| Data Error | A packet was received from a link with the EP bit set. | DataEn, McaUsPwDatErrEn |
| Protocol Error | Protocol error detected. | ProtEn |
| DEV Error | SVM DEV error detected. | DevErrEn |

**Table 118: NB error signatures, part 1**

| **Error Type** | **20:16 Ext. Error** | **Error Code (see D18F3x48 for encoding)** | | | | | |
|---|---|---|---|---|---|---|---|
| | | **Type** | **10:9 PP** | **8 T** | **7:4 RRRR** | **3:2 II/TT** | **1:0 LL** |
| Reserved. | 0_0000 | - | - | - | - | - | - |
| Reserved | 0_0001 | | | | | | |
| Sync Error | 0_0010 | BUS | OBS | 0 | GEN | GEN | LG |
| Mst Abort | 0_0011 | BUS | SRC/OBS | 0 | RD/WR | MEM/IO | LG |
| Tgt Abort | 0_0100 | BUS | SRC/OBS | 0 | RD/WR | MEM/IO | LG |
| WDT Error | 0_0111 | BUS | GEN | 1 | GEN | GEN | LG |
| Data Error | 0_1010 | BUS | SRC/OBS | 0 | RD/WR/ DWR | MEM/IO | LG |
| Protocol Error | 0_1011 | BUS | OBS | 0 | GEN | GEN | LG |
| DEV Error | 0_1001 | BUS | SRC/OBS | 0 | RD/WR | MEM/IO | LG |

**Table 119: NB error signatures, part 2**

| Error Type | D18F3x4C settings | | | |
|---|---|---|---|---|
| | UC | AddrV | PCC | ErrCPU |
| Sync Error | 1 | 0 | 1 | - |
| Mst Abort | 1 | 1 | If CPU source | Y |
| Tgt Abort | 1 | 1 | If CPU source | Y |
| WDT Error | 1 | 1 | 1 | - |
| DEV Error | 1 | 1 | 0 | - |

### D18F3x4C MCA NB Status High Register

Cold reset: xxxx_xxxxh.

Software is normally only allowed to write 0's to this register to clear the fields so subsequent errors may be logged. See MSRC001_0015[McStatusWrEn]. This register may be accessed through [The NB Machine Check Status Register (MC4_STATUS)] MSR0000_0411 as well.

| Bits | Description |
|---|---|
| 31 | **Val: error valid**. Read-write; set-by-hardware. 1=This bit indicates that a valid error has been detected. This bit should be cleared to 0 by software after the register has been read. |
| 30 | **Over: error overflow**. Read-write; set-by-hardware. 1=The NB attempted to record a new error with Val already set; an overflow occurred and the new error was not written. When Over and UC are both set, critical error information may have been lost, and software should terminate system processing to prevent data corruption (see 2.16.3 [Handling Machine Check Exceptions]). For certain conditions, a new error seen while Val is set may cause Over to be set, regardless of error priority or whether information was lost. Therefore, if UC is not indicated, there is no need to terminate the system, as any lost information was not critical.<br><br>• If the existing error is overwritten, Over is not set.<br>• Table 42 describes the conditions under which a younger error overwrites an older error. |
| 29 | **UC: error uncorrected**. Read-write; set-by-hardware. 1=The error was not corrected by hardware. |
| 28 | **En: error enable**. Read-write; set-by-hardware. 1=The MCA error reporting is enabled for this error in the MCA Control Register. |
| 27 | Reserved. |
| 26 | **AddrV: error address valid**. Read-write; set-by-hardware. 1=The address saved in the address register is the address where the error occurred. |
| 25 | **PCC: processor context corrupt**. Read-write; set-by-hardware. 1=The state of the processor may be corrupted by the error condition. Reliable restarting might not be possible. |
| 24:5 | Reserved. |

| | |
|---|---|
| 4 | **BusErr: bus error**. Read-write; set-by-hardware. 1=The error was is associated with a transaction to or from the root complex. |
| 3:0 | **ErrCPU: error associated with core N**. Read-write; set-by-hardware. This field indicates which core within the processor is associated with the error.<br>ErrCPU[3] = Error associated with core 3.<br>ErrCPU[2] = Error associated with core 2.<br>ErrCPU[1] = Error associated with core 1.<br>ErrCPU[0] = Error associated with core 0. |

### D18F3x50 MCA NB Address Low Register

Cold reset: xxxx_xxxxh.

[The MCA NB Address Low Register] D18F3x50 and [The MCA NB Address High Register] D18F3x54 carry the address associated with a machine check error, other fields, or both.

IF (D18F3x48[ErrorCodeExt] == 07h) THEN

| Bits | Description |
|---|---|
| 31:2 | **ErrorAddr[31:2]**. Read-write. Error address. |
| 1 | **RspDispatched**. Read-write. 1=Response from link was dispatched. |
| 0 | **ReqDispatched**. Read-write. 1=Request was dispatched. |

ELSE

| Bits | Description |
|---|---|
| 31:0 | **ErrorAddr[31:0]**. Read-write. Error address. |

ENDIF

### D18F3x54 MCA NB Address High Register

Cold reset: xxxx_xxxxh.

IF (D18F3x48[ErrorCodeExt] == 07h) THEN

| Bits | Description |
|---|---|
| 31:28 | **WaitCode: RAQ wait code**. Read-write.<br>• [63]=1 means all inbound data has not been transferred.<br>• [62]=1 means ordering rules not satisfied for Dispatch of Response.<br>• [61]: Reserved.<br>• [60]=1 means lack of downstream credits. |
| 27 | **Priority**. 1=High. 0=Low. |

| Bits | Description |
|---|---|
| 26:24 | **PktRequester: packet requester**. Read-write.<br>Bits: Requester<br>000b: Core 0<br>001b: Core 1<br>010b: Core 2<br>011b: Core 3<br>100b: IO device<br>101b: NB 11xb<br>Reserved |
| 23:22 | **PktTarget: packet target**. Read-write.<br>Bits: Target<br>00b: DRAM<br>01b: NB<br>10b: MMIO/IO/FCH<br>11b: Reserved |
| 21:16 | **LinkCmd: link command**. Read-write. |
| 15:8 | Reserved. |
| 7:0 | **ErrorAddr[39:32]: error address**. Read-write. |

ELSE

| Bits | Description |
|---|---|
| 31:8 | Reserved. |
| 7:0 | **ErrorAddr[39:32]: error address**. Read-write. |

ENDIF

### D18F3x64 Hardware Thermal Control (HTC) Register

See 2.10.3.1 [PROCHOT_L and Hardware Thermal Control (HTC)] for information on HTC. D18F3x64 is reserved if D18F3xE8[HtcCapable]=0.

| Bits | Description |
|---|---|
| 31 | **HtcLock: HTC lock**. Read; write-1-only. Reset: 0. SBIOS: 1. 1=HtcPstateLimit, HtcHystLmt, HtcTmpLmt, and HtcEn are read-only. 0=HtcPstateLimit, HtcHystLmt, HtcTmpLmt, and HtcEn are read-write. |
| 30:28 | **HtcPstateLimit: HTC P-state limit select**. Read-write. Reset: Product-specific. Specifies the P-state limit of all cores when in the HTC-active state. The HtcPstateLimit to apply is not changed if the value of this field is greater than MSRC001_0061[PstateMaxVal]. See 2.10.3.1 [PROCHOT_L and Hardware Thermal Control (HTC)]. This field uses hardware P-state numbering, see 2.5.3.1.2.2 [Hardware P-state Numbering]. |
| 27:24 | **HtcHystLmt: HTC hysteresis**. Read-write. Reset: Product-specific. The processor exits the HTC-active state when Tctl is less than HTC temperature limit (HtcTmpLmt) minus HTC hysteresis (HtcHystLmt).<br>Bits: Description<br>Fh-0h: <HtcHystLmt*0.5> |

| | |
|---|---|
| 23 | **HtcSlewSel: HTC slew-controlled temperature select**. Read-write. Reset: 0. 1=HTC logic is driven by the slew-controlled temperature, Tctl, specified in [The Reported Temperature Control Register] D18F3xA4. 0=HTC logic is driven by the measured control temperature with no slew controls. |
| 22:16 | **HtcTmpLmt: HTC temperature limit**. Read-write. Read-write. Reset: Product-specific. The processor enters the HTC-active state when Tctl reaches or exceeds the temperature limit defined by this register.<br>Bits: 7Fh-00h; Description: <HtcTmpLmt*0.5 + 52> |
| 15:8 | Reserved. |
| 7 | **PslApicLoEn: P-state limit lower value change APIC interrupt enable**. Read-write. Reset: 0. PslApicLoEn and PslApicHiEn enable interrupts using [The Thermal Local Vector Table Entry] APIC330 of each core when the active P-state limit in [The P-State Current Limit Register] MSRC001_0061[CurPstateLimit] changes. PslApicLoEn enables the interrupt when the limit value becomes lower (indicating higher performance). PslApicHiEn enables the interrupt when the limit value becomes higher (indicating lower performance). 1=Enable interrupt. |
| 6 | **PslApicHiEn: P-state limit higher value change APIC interrupt enable**. Read-write. Reset: 0. See: PslApicLoEn. |
| 5 | **HtcActSts: HTC-active status**. Read; set-by-hardware; write-1-to-clear. Reset: 0. This bit is set by hardware when the processor enters the HTC-active state. It is cleared by writing a 1 to it. |
| 4 | **HtcAct: HTC-active state**. Read-only. Reset: X. 1=The processor is currently in the HTC-active state. 0=The processor is not in the HTC-active state. |
| 3:1 | Reserved. |
| 0 | **HtcEn: HTC enable**. Read-write. Reset: 0. BIOS: IF (D18F3x64[HtcTmpLmt]==0) THEN 0 ELSE 1 ENDIF. 1=HTC is enabled; the processor is capable of entering the HTC-active state. |

### D18F3x6C Upstream Data Buffer Count Register

Read-write. BIOS: Table 8.

- Buffer allocation requirements:
  - UpHiRespDBC + UpHiNpreqDBC + UpHipreqDBC + UpLoRespDBC + UpLoNpreqDBC + UpLoPreqDBC <=16.
  - One buffer must be allocated for each enabled channel. The low priority channels are always enabled.
  - Buffer allocations cannot be decreased through software. To decrement the buffers allocated to a channel generate a reset and then reassign buffers to the new settings.

| Bits | Description |
|---|---|
| 31:28 | Reserved. |
| 27:24 | **UpHiRespDBC: upstream high priority response data buffer count**. Read-write. Reset: 0h. |
| 23:20 | **UpHiNpreqDBC: upstream high priority non-posted request data buffer count**. Read-write. Reset: 0h. |
| 19:16 | **UpHiPreqDBC: upstream high priority posted request data buffer count**. Read-write. Reset: 0h. |
| 15:12 | Reserved. |
| 11:8 | **UpLoRespDBC: upstream low priority response data buffer count**. Read-write. Reset: 1h. |
| 7:4 | **UpLoNpreqDBC: upstream low priority non-posted request data buffer count**. Read-write. Reset: 1h. |

| | |
|---|---|
| 3:0 | **UpLoPreqDBC: upstream low priority posted request data buffer count**. Read-write. Reset: 1h. |

### D18F3x74 Upstream Command Buffer Count Register

Read-write. BIOS: Table 8.

- Buffer allocation requirements:
  - UpHiNpreqCBC + UpHiPreqCBC + UpLoNpreqCBC + UpLoPreqCBC <=16.
  - UpLoRespCBC <=8.
  - One buffer must be allocated for each enabled channel. The low priority channels are always enabled.
  - Buffer allocations cannot be decreased through software. To decrement the buffers allocated to a channel generate a reset and then reassign buffers to the new settings.

| Bits | Description |
|---|---|
| 31:28 | Reserved. |
| 27:24 | **UpHiRespCBC: upstream high priority response command buffer count**. Reset: 0. |
| 23:20 | **UpHiNpreqCBC: upstream high priority non-posted command buffer count**. Reset: 0. |
| 19:16 | **UpHiPreqCBC: upstream high priority posted command buffer count**. Reset: 0. |
| 15:12 | Reserved. |
| 11:8 | **UpLoRespCBC: upstream low priority response command buffer count**. Reset: 1h. |
| 7:4 | **UpLoNpreqCBC: upstream low priority non-posted command buffer count**. Reset: 1h. |
| 3:0 | **UpLoPreqCBC: upstream low priority posted command buffer count**. Reset: 1h. |

### D18F3x7C In-Flight Queue Buffer Allocation Register

Read-write. BIOS: 2.9.3.5.

- Buffer allocation requirements:
  - FreePoolBC + D18F3x17C[HiPriNpBC] + D18F3x17C[HiPriPBC] + LoPriNpBC + LoPriPBC + CpuBC <= 28
  - One buffer must be allocated for each enabled channel. The CPU and low priority channels are always enabled.
  - Buffer allocations cannot be decreased through software. To decrement the buffers allocated to a channel generate a reset and then reassign buffers to the new settings.

| Bits | Description |
|---|---|
| 31:30 | Reserved. |
| 29:24 | **FreePoolBC: free pool buffer count**. Reset: 1h. This field must be at least 1. |
| 23:22 | Reserved. |
| 21:16 | **LoPriNpBC: low priority channel non-posted buffer count**. Reset: 1h. |
| 15:14 | Reserved. |
| 13:8 | **LoPriPBC: low priority channel posted buffer count**. Reset: 1h. |
| 7:6 | Reserved. |
| 5:0 | **CpuBC: CPU buffer count**. Reset: 1h. |

### D18F3x80 ACPI Power State Control Low

Read-write. Reset: 0000_0000h. BIOS: 0000_0000h.

D18F3x80 and D18F3x84 consist of eight identical 8-bit registers, one for each System Management Action Field (SMAF) code associated with STPCLK assertion commands from the link. Refer to the table below for the associated ACPI state and SMAF code for each of the 8 registers. Some ACPI states and associated SMAF codes may not be supported in certain conditions. Refer to Table 4 for information on which states are supported. See D18F4x138 and D18F4x13C for the clock divisors that apply to each SMAF code.

**Table 120: SMAF to ACPI state mapping**

| SMAF | ACPI State | Description |
|---|---|---|
| 7 | Reserved. | Reserved. |
| 6 | S4/S5 | Initiated by a processor access to the ACPI-defined PM1_CNTa register. |
| 5 | Reserved. | Reserved. |
| 4 | S3 | Initiated by a processor access to the ACPI-defined PM1_CNTa register. |
| 3 | Reserved. | Reserved. |
| 2 | Reserved. | Reserved. |
| 1 | Reserved. | Reserved. |
| 0 | Reserved. | Reserved. |

| Bits | Description |
|---|---|
| 31:0 | Reserved. |

### D18F3x84 ACPI Power State Control High

Reset: 0000_0000h. Read-write. BIOS: 0006 0006h.

See D18F3x80 for detail about the System Management Action Field (SMAF).

| Bits | Description |
|---|---|
| 31:19 | Reserved. |
| 18 | **Smaf6DramMemClkTri: SMAF 6 DRAM memory clock tristated**. BIOS: 1. See: D18F3x84[Smaf4DramMemClkTri]. |
| 17 | **Smaf6DramSr: SMAF 6 DRAM self-refresh**. BIOS: 1. See: D18F3x84[Smaf4DramSr]. |
| 16:3 | Reserved. |
| 2 | **Smaf4DramMemClkTri: SMAF 4 DRAM memory clock tristated**. 1=MEMCLKs are tristated while in the low-power state. DramSr is required to be set if this bit is set. See 2.5.6.1 [DRAM Self-Refresh]. |
| 1 | **Smaf4DramSr: SMAF 4 DRAM self-refresh**. 1=DRAM is enabled to be placed into self-refresh while in the low-power state. See 2.5.6.1 [DRAM Self-Refresh]. |
| 0 | Reserved. |

### D18F3x88 NB Configuration Low Registers

Reset: 0000_0200h. MSRC001_001F[31:0] is an alias of D18F3x88.

| Bits | Description |
|---|---|
| 31:0 | Reserved. |

### D18F3x8C NB Configuration High Registers

Reset: 0000_0000h. MSRC001_001F[63:32] is an alias of D18F3x88.

| Bits | Description |
|---|---|
| 31:27 | Reserved. |
| 26 | **EnConvertToNonIsoc**: **enable conversion to non-isochronous**. Read-write. BIOS: 1. 1=Convert peer-to-peer isochronous requests to non-isochronous requests (the Isoc bit in the downstream request packet is low); however, the Isoc bit in the downstream response to the requester is still set in such a case. In non-IFCM mode, the link-defined Isoc bit in the request packet is cleared as it is reflected downstream in a peer-to-peer access as well. |
| 25:15 | Reserved. |
| 14 | **EnableCf8ExtCfg: enable CF8 extended configuration cycles**. Read-write. 1=Allows the IO configuration space access method, IOCF8 and IOCFC, to be used to generate extended configuration cycles by enabling IOCF8[27:24]. |
| 13 | **DisUsSysMgtReqToNcHt: disable upstream system management request to link**. Read-write. 1=Disables downstream reflection of upstream STPCLK and x86 legacy input system management commands (in order to work around potential deadlock scenarios related to reflection regions). |
| 12:0 | Reserved. |

### D18F3xA0 Power Control Miscellaneous Register

| Bits | Description |
|---|---|
| 31 | **CofVidProg: COF and VID of P-states programmed**. Read-only. Reset: Product-specific. |
| 30:28 | Reserved. |
| 27:16 | **ConfigId: configuration identifier**. Read-only. Reset: Product-specific. This field specifies the configuration ID associated with the product. |
| 15:10 | Reserved. |
| 9 | **SviHighFreqSel: SVI high frequency select**. Read-write. Reset: 0. BIOS: 1. 0=400 kHz. 1=3.4 MHz. Writes to this field take effect after the next SVI command. |
| 8 | Reserved. |
| 7 | **PsiVidEn: PSI_L bit VID enable**. Read-write. Cold reset: 0. BIOS: See 2.5.1.4.1 [PSI_L Bit]. This bit specifies how the PSI_L bit for VDD is controlled. 1=See D18F3xA0[PsiVid]. 0=The PSI_L bit is always 1. |
| 6:0 | **PsiVid: PSI_L bit VID threshold**. Read-write. Cold reset: 0. BIOS: See 2.5.1.4.1 [PSI_L Bit]. If D18F3xA0[PsiVidEn]==1, this field specifies a VID code that determines the state of the PSI_L bit for the VDD plane. Whenever the VID code generated by the processor for the VDD plane is less than (voltage is greater than) PsiVid, the PSI_L bit is sent as a 1. Whenever the VID code generated by the processor for the VDD plane is greater than or equal to (voltage less than or equal to) PsiVid, the PSI_L bit is sent as a 0. See 2.5.1.4.1 [PSI_L Bit]. |

### D18F3xA4 Reported Temperature Control Register

The slew rate controls in this register are used to filter processor temperature measurements. Separate controls are provided for a measured temperature that is higher or lower than Tctl. The per-step timer counts as long as the measured temperature stays either above or below Tctl. Each time the measured temperature changes to the other side of Tctl, the step timer resets, and Tctl is not changed. If, for example, step times are enabled in both directions, Tctl=62.625, and the measured temperature keeps jumping quickly between 62.5 and 63.0, then (assuming the step times are long enough) Tctl would not change. However, once the measured temperature settles on one side of Tctl, Tctl can step toward the measured temperature. If the difference of measured temperature minus Tctl is greater than the value set by MaxTmpDiffUp, then Tctl is set equal to the measured temperature. See 2.10 [Thermal Functions].

| Bits | Description |
|---|---|
| 31:21 | **CurTmp: current temperature**. Read-only. Reset: X. Specifies the current control temperature with the slew-rate controls applied. See 2.10.1 [The Tctl Temperature Scale].<br>Bits: Definition<br>7FFh-000h: <CurTmp*0.125>. |
| 20:13 | Reserved. |
| 12:8 | **PerStepTimeDn: per 1/8th step time down**. Read-write. Cold reset: 18h. BIOS: 0Fh. This specifies the time per 1/8th step of Tctl when the measured temperature is less than the Tctl. It is encoded the same as PerStepTimeUp. |
| 7 | **TmpSlewDnEn: temperature slew downward enable**. Read-write. Cold reset: 0b. BIOS: 1b. 1=Slew rate controls in the downward direction are enabled. 0=Downward slewing disabled; if the measured temperature is detected to be less than Tctl then Tctl is updated to match the measured temperature. |
| 6:5 | **TmpMaxDiffUp: temperature maximum difference up**. Read-write. Cold reset: 00b. BIOS: 11b. This specifies the maximum difference between Tctl and the measured temperature, when the measured value is greater than Tctl (i.e., when the temperature has risen). If this difference exceeds the specified value, Tctl jumps to the measured temperature value. This field is encoded as follows:<br>Bits: Definition<br>00b: Upward slewing disabled; if the measured temperature is detected to be greater than Tctl then Tctl is updated to match the measured temperature.<br>01b: Tctl is held to less than or equal to measured temperature minus 1.0.<br>10b: Tctl is held to less than or equal to measured temperature minus 3.0.<br>11b: Tctl is held to less than or equal to measured temperature minus 9.0. |
| 4:0 | **PerStepTimeUp: per 1/8th degree step time up**. Read-write. Cold reset: 00h. BIOS: 0Fh. This specifies the time per 1/8-degree step of Tctl when the measured temperature is greater than the reported temperature. It is encoded as follows:<br>Bits: Definition<br>1Fh-00h: <(PerStepTimeUp[2:0] + 1) * 10^PerStepTimeUp[4:3]> ms, ranging from 1 ms to 8000 ms. |

### D18F3xD4 Clock Power/Timing Control 0 Register

| Bits | Description |
|---|---|
| 31:19 | Reserved. |
| 18 | Reserved. |

| | |
|---|---|
| 17 | **ClockGatingEnDram: clock gating enable DRAM**. Read-write. Reset: 0. BIOS: 1. Specifies whether NCLK gating to the DRAM controller is enabled. 1=Enabled. 0=Disabled. See 2.5.4.3 [NB Clock Gating]. |
| 16 | **DisNclkGatingIdle: disable NCLK gating when idle**. Read-write. Reset: 0. 1=NCLK gating is disallowed when NCLK is ramped down. 0=NCLK gating is allowed when NCLK is ramped down. See 2.5.4.2 [NB Clock Ramping] and 2.5.4.3 [NB Clock Gating]. |
| 15:12 | **NbOutHyst: northbridge outbound hysteresis**. Read-write. Reset: 0. BIOS: 04h. Specifies the hysteresis time after the IFQ is emptied until the NB deasserts the outbound wake signal.<br>Bits — Time<br>0h — 0<br>Eh-1h — hysteresis time = 80ns * 2^(OnionOutHyst - 1)<br>Fh — wake deassertion disabled<br>See 2.5.4.3 [NB Clock Gating]. |
| 11:8 | **ClkRampHystSel: clock ramp hysteresis select**. Read-write. Reset: 0. BIOS: 0Fh. Specifies the hysteresis time used when ramping up to service probe requests. Hysteresis time= 320ns * (1 + ClkRampHystSel). See 2.5.3.2.5 [C-states and Probe Requests]. |
| 7 | **ShallowHaltDidAllow**. Read-write. Reset: 0. BIOS: 1. See D18F4x1A8[SingleHaltCpuDid]. |
| 6 | Reserved. |
| 5:0 | **MainPllOpFreqId: main PLL operating frequency ID**. Read-write; reset-applied. Cold reset: Product-specific. Specifies the COF of the main PLL.<br>• Main PLL COF = 100 MHz * (D18F3xD4[MainPllOpFreqId] + 10h).<br>• D18F3xD4[MainPllOpFreqId] must be programmed as specified by MSRC001_0071[MainPllOpFreqIdMax]. |

**D18F3xD8 Clock Power/Timing Control 1 Register**

| Bits | Description |
|---|---|
| 31:29 | Reserved. |
| 28:12 | Reserved. |
| 11:7 | **ExtndTriDly: extend tristate delay**. Read-write. Cold reset: 0_1111b. Specifies a delay in REFCLKs that the processor leaves the SVD signal tristated after receiving an ACK from the voltage regulator. |

| | |
|---|---|
| 6:4 | **VSRampSlamTime**. Read-write. Reset: 0. BIOS: Voltage Ramp Time[1]. Specifies the time the processor waits for voltage increases to complete before beginning an additional voltage change or a frequency change. See 2.5.1.5.1 [Hardware-Initiated Voltage Transitions].<br><br>Ramp time = (VSRampSlamTime / 12.5mV) * ABS(destination voltage - current voltage).<br><br>Bits / Time / Bits / Time<br>000b 6.25 microseconds 100b 2.50 microseconds<br>001b 5.00 microseconds 101b 1.67 microseconds<br>010b 4.17 microseconds 110b 1.25 microseconds<br>011b 3.13 microseconds 111b 1.00 microseconds<br><br>1. Voltage Ramp Time = maximum time to change VDD or VDDNB by 12.5mV rounded to the next higher encoding.<br><br>For example, if the VDD regulator slew rate is 8mV/us and the VDDNB regulator slew rate is 5.5mV/us, it takes the VDD regulator 1.56us to change 12.5mV and it takes the VDDNB regulator 2.27us to change 12.5mV. In this case, BIOS should set this field to 2.5us. |
| 3:0 | Reserved. |

**D18F3xDC Clock Power/Timing Control 2 Register**

| Bits | Description |
|---|---|
| 31 | **CnbCifClockGateEn: NB CIF clock gating enable**. Read-write. Reset: 0. BIOS: 1. Specifies whether dynamic clock gating on the NBCIF is enabled. 1=Enabled. 0=Disabled. See 2.5.4.3 [NB Clock Gating]. |
| 30 | **NbClockGateEn: northbridge clock gating enable**. Read-write. Reset: 0. BIOS: 0. Specifies whether dynamic clock gating on the NB is enabled. 1=Enabled. 0=Disabled. See 2.5.4.3 [NB Clock Gating]. |
| 29:27 | **NbClockGateHyst: northbridge clock gating hysteresis**. Read-write. Reset: 0. BIOS: 011b. Specifies how long hardware waits after the IFQ is empty and all cores are in a non-C0 state before gating the NB clocks.<br>Bits / Time / Bits / Time<br>000b 0 100b 3us<br>001b 320ns 101b 5us<br>010b 500ns 110b 10us<br>011b 1us 111b 20us<br>See 2.5.4.3 [NB Clock Gating]. |

| | |
|---|---|
| 26:20 | **NbPs0NclkDiv: NCLK divisor**. Read-write. Reset: Product-specific. BIOS: 2.5.4.1.1. Specifies the NCLK divisor when in NBP0.<br><br>• The clock divisor can be calculated using the following table:<br>Bits — Resulting Clock Divisor<br>07h-00h — Reserved<br>3Fh-08h — (NbPs0NclkDiv * 0.25)<br>7Fh-40h — Reserved.<br>• 50% clock duty cycles are obtained at all divisors integer and half-integer divisors only. For example, /2, /2.5, /3.0, and /3.5 give 50% clock duty cycles whereas /3.25 does not.<br>• Divisor examples:<br>  • Example #1: If D18F3xDC[NbPs0NclkDiv] = 0Ch = 12d, then the NCLK divisor = 12 * 0.25 = /3.0<br>• The NCLK COF can be calculated using the following equation:<br>  • COF = (main PLL frequency specified by D18F3xD4[MainPllOpFreqId]) / clock divisor.<br>  • Example: If D18F3xD4[MainPllOpFreqId] = 10h = 3.2GHz and D18F3xDC[NbPs0NclkDiv] = 3Eh = /15.5, then the NCLK COF = 206.45MHz.<br>• Writes that change the value of this field cause NCLK to transition to the new divisor if the processor is currently in NBP0. This occurs regardless of the state of D18F6x90[NbPsCap].<br>• Software may only change the value of this field if either:<br>  • The sequence described in 2.9.3 [DCT/DRAM Initialization and Resume] has not been run, or<br>  • DRAM has been placed into self-refresh. See D18F2x[1,0]90[EnterSelfRef].<br><br>See 2.5.4.1 [NB P-states]. |
| 19 | **NclkFreqDone: NCLK frequency change done**. Read-only. Reset: 0. 1=NCLK frequency change complete. 0=NCLK frequency change in progress. |
| 18:12 | **NbPs0Vid**: **NB VID**. Read-write. Reset: Product-specific. BIOS: 2.5.4.1.1. Specifies the VID for VDDNB when in NBP0. Writes to this field cause the VID being output for VDDNB to change if the processor is currently in NBP0. This occurs regardless of the state of D18F6x90[NbPsCap]. See the AMD Voltage Regulator Specification, #40182 for encodings. See 2.5.4.1 [NB P-states].<br><br>Writing this field while D18F6x90[NbPsCtrlDis] == 0 may result in undefined behavior. Whenever this field is written, software must wait the RampTime specified by D18F3xD8[VSRampSlamTime] before clearing D18F6x90[NbPsCtrlDis] to 0, changing the value of D18F3xDC[NbPs0NclkDiv], or changing the value of D18F6x90[NbPsForceSel]. |
| 11 | Reserved. |
| 10:8 | **HwPstateMaxVal: P-state maximum value**. Read-write. Reset: specified by the reset state of MSRC001_00[6B:64][PstateEn]; the reset value is the highest P-state number corresponding to the MSR in which PstateEn is set (e.g., if MSRC001_0064 and MSRC001_0065 have this bit set and the others do not, then HwPstateMaxVal=1; if PstateEn is not set in any of these MSRs, then HwPstateMaxVal=0). This specifies the highest P-state value (lowest performance state) supported by the hardware. See MSRC001_0061[PstateMaxVal]. This field uses hardware P-state numbering. See 2.5.3.1.2.2 [Hardware P-state Numbering]. |
| 7:0 | Reserved. |

**D18F3xE4 Thermtrip Status Register**

| Bits | Description |
|---|---|
| 31 | **SwThermtp: software THERMTRIP**. Write-1-only. Reset: 0. Writing a 1 to this bit position induces a THERMTRIP event. This bit returns 0 when read. This is a diagnostic bit, and it should be used for testing purposes only. |
| 30:8 | Reserved. Reads return an undefined value. |
| 7:6 | Reserved. |
| 5 | **ThermtpEn: THERMTRIP enable**. Read-only. Reset: Product-specific. 1=The THERMTRIP state as specified in section 2.10.3.3 [THERMTRIP] is supported by the processor. |
| 4 | Reserved. |
| 3 | **ThermtpSense: THERMTRIP sense**. Read-only. Cold-reset: 0. 1=The processor temperature exceeded the THERMTRIP value (regardless as to whether the THERMTRIP state (ThermtpEn) is enabled). This bit is also set when the diagnostic bit SwThermtp = 1. |
| 2 | Reserved. |
| 1 | **Thermtp: THERMTRIP**. Read-only. Reset: 0. 1=The processor has entered the THERMTRIP state. |
| 0 | Reserved. |

**D18F3xE8 Northbridge Capabilities Register**

Read-only. Unless otherwise specified, 1=The feature is supported by the processor; 0=The feature is not supported.

| Bits | Description |
|---|---|
| 31:29 | Reserved. |
| 28 | **LHtcCapable: LHTC capable**. Reset: Product-specific. |
| 27:14 | Reserved. |
| 13:12 | **CmpCap: CMP capable**. Reset: Product-specific. Specifies the number of cores enabled on the device.<br>Bits / Definition / Bits / Definition<br>00b / 1 / 10b / 3<br>01b / 2 / 11b / 4 |
| 11 | Reserved. |
| 10 | **HtcCapable: HTC capable**. Reset: Product-specific. |
| 9 | **SvmCapable: SVM capable**. Reset: Product-specific. |
| 8 | **MctCap: memory controller (on the processor) capable**. Reset: 1. |
| 7:5 | **DdrMaxRate**. Reset: Product-specific. Specifies the maximum DRAM data rate that the processor is designed to support.<br>Bits / DDR limit / Bits / DDR limit<br>000b / No limit / 100b / 800 MT/s<br>001b / 1600 MT/s / 101b / 667 MT/s<br>010b / 1333 MT/s / 110b / 533 MT/s<br>011b / 1067 MT/s / 111b / 400 MT/s |

| | |
|---|---|
| 4:1 | Reserved. |
| 0 | **DctDualCap: two-channel DRAM capable**. Reset:1b. |

### D18F3xF0 DEV Capability Header Register

See 2.8.3 [DMA Exclusion Vectors (DEV)]. DMA Exclusion Vectors (DEV) are contiguous arrays of bits in physical memory. There is no support for MMIO DEV tables. Each bit in the DEV table represents a 4KB page of physical memory; the DEV applies to accesses that target system memory and MMIO. The DEV table is packed as follows: bit[0] of byte 0 (pointed to by the DEV table base address, D18F3xF8_x0 and D18F3xF8_x1) controls the first 4K bytes of physical memory (starting at address 00_0000_0000h); bit[1] of byte 0 controls the second 4K bytes of physical memory; etc. When a DEV table bit is set to one, accesses to that physical page by external DMA devices is not allowed. If an external device attempts to access a protected physical page, then the processor master aborts the request.

In addition, the processor supports multiple protection domains. There is a DEV table for each protection domain. Link-defined UnitIDs or RequesterID's may be assigned to the DEV of a specific protection domain through D18F3xF8_x2. DEV table walks for each protection domain are cached in the NB to reduce the number DEV table access to system memory.

The DEV function is configured through D18F3xF0, D18F3xF4, D18F3xF8, and an array of registers called F3xF8_DF[7:0], which are defined following D18F3xF8. [The DEV Function/Index Register] D18F3xF4 and [The DEV Data Port] D18F3xF8 are used to access F3xF8_DF[7:0]. The register number (i.e., the number that follows "_DF" in the register mnemonic) is specified by D18F3xF4[DevFunction]. In addition, D18F3xF8_x0, D18F3xF8_x1, and D18F3xF8_x2 are each instantiated multiple times, indexed by D18F3xF4[DevIndex]. Access to these registers is accomplished as follows:

- Reads:
  - Write the register number to D18F3xF4[DevFunction, DevIndex].
  - Read the register contents from D18F3xF8.

- Writes:
  - Write the register number to D18F3xF4[DevFunction, DevIndex].
  - Write the register contents to D18F3xF8.

IF (D18F3xE8[SvmCapable] == 0) THEN

| Bits | Description |
|---|---|
| 31:0 | Reserved. |

ELSE

| Bits | Description |
|---|---|
| 31:22 | Reserved. |
| 21 | **IntCap: interrupt reporting capability**. Read-only. Reset: 0. 0=Interrupt reporting of DEV protection violations is not present on this device. |
| 20 | **MceCap: MCE reporting capability**. Read-only. Reset: 1. Indicates that machine check architecture reporting of DEV protection violations is present on this device. |
| 19 | Reserved. |

| | |
|---|---|
| 18:16 | **CapType: DEV capability block type**. Read-only. Reset: 000b. Specifies the layout of the Capability Block. |
| 15:8 | **CapPtr: capability pointer**. Read-only. Reset: 00h. Indicates that this is the last capability block. |
| 7:0 | **CapId: capability ID**. Read-only. Reset: 0Fh. Indicates a DEV capability block. |

ENDIF.

### D18F3xF4 DEV Function/Index Register

Reset: 0000_0000h.
IF (D18F3xE8[SvmCapable] == 0) THEN

| Bits | Description |
|---|---|
| 31:0 | Reserved. |

ELSE

| Bits | Description |
|---|---|
| 31:16 | Reserved. |
| 15:8 | **DevFunction**. Read-write. See D18F3xF0. Valid values for this field are 00h through 07h. Writing invalid values may result in undefined behavior. |
| 7:0 | **DevIndex**. Read-write. See D18F3xF0. Valid values for this field are (1) 00h through (D18F3xF8_x3[NDomains] - 1) when either D18F3xF8_x0 or D18F3xF8_x1 are being accessed and (2) 00h through (D18F3xF8_x3[NMaps] + D18F3xF8_x3[NSrcMaps] - 1) when D18F3xF8_x2 is being accessed; this field is reserved for accesses to all other DEV configuration registers. Writing invalid values may result in undefined behavior. |

ENDIF.

### D18F3xF8 DEV Data Port

Reset: 0000_0000h. See D18F3xF0 for details about this port.

IF (D18F3xE8[SvmCapable] == 0) THEN

| Bits | Description |
|---|---|
| 31:0 | Reserved. |

ELSE

| Bits | Description |
|---|---|
| 31:0 | **DevData**. Read-write. See D18F3xF8_x0, D18F3xF8_x1, D18F3xF8_x2, D18F3xF8_x3, D18F3xF8_x4, D18F3xF8_x5, D18F3xF8_x6, and D18F3xF8_x7. |

ENDIF.

### D18F3xF8_x0 DEV Base Address/Limit Low Register

Reset: 0000_0000h.

This register is instantiated multiple times, specified by D18F3xF8_x3[NDomains]. Each instantiation corresponds to a protection domain number, identical to D18F3xF4[DevIndex], which is the index to the instantia-

tion. See D18F3xF0.

| Bits | Description |
|---|---|
| 31:12 | **BaseAddress[31:12]: DEV table base address bits[31:12]**. Read-write. These bits are combined with D18F3xF8_x1[BaseAddress[39:32]] to specify the base address of the DEV table. The DEV table is required to be in either non-cacheable or write-through memory. Placing DEV tables in MMIO space is not supported. If any part of the DEV table is in other than system memory, then undefined behavior results. |
| 11:7 | Reserved. |
| 6:2 | **Size: DEV table size**. Read-write. These bits specify the size of the memory region that the DEV table covers. The corresponding DEV table size is 128KB*(2^Size).<br>Bits Definition<br>08h-00h <4*2^Size> GB<br>1Fh-09h Reserved. |
| 1 | **Protect: protect out-of-range addresses**. Read-write. 0=DMA accesses to addresses that are outside the range covered by the DEV table are allowed. 1=DMA accesses to addresses that are outside the range covered by the DEV table are protected. |
| 0 | **Valid: DEV table valid**. Read-write. 1=The DEV table for the protection domain specified by D18F3xF4[DevIndex] is enabled. 0=The DEV table is not enabled; all IO accesses from devices assigned to the corresponding protection domain are allowed. |

### D18F3xF8_x1 DEV Base Address/Limit High Register

Reset: 0000_0000h.

This register is instantiated multiple times, specified by D18F3xF8_x3[NDomains]. Each instantiation corresponds to a protection domain number, identical to D18F3xF4[DevIndex], which is the index to the instantiation. See D18F3xF0.

| Bits | Description |
|---|---|
| 31:8 | Reserved. |
| 7:0 | **BaseAddress[39:32]: DEV table base address bits[39:32]**. Read-write. See D18F3xF8_x0[BaseAddress]. |

### D18F3xF8_x2 DEV Map Register

Reset: 0000_0000h.

The DEV Map register maps internal unit IDs (see Table 121) to DEV protection domains. This register is instantiated the number of times specified D18F3xF8_x3[NMaps].

**Table 121: Internal unit ID mapping**

| Unit ID | Internal Device |
|---|---|
| 01h | GPU |
| 02h | Root Port (Bus 0, Device 2) |
| 03h | Root Port (Bus 0, Device 3) |
| 04h | Root Port (Bus 0, Device 4) |

**Table 121: Internal unit ID mapping**

| Unit ID | Internal Device |
|---|---|
| 05h | Root Port (Bus 0, Device 5) |
| 06h | Root Port (Bus 0, Device 6) |
| 07h | Root Port (Bus 0, Device 7) |
| 08h | FCH |
| 1Fh:09h | Reserved |

If Valid[x] is set, then the address of DMA requests received by the processor from an internal device with a UnitID of Unit[x] are checked against the DEV table of protection domain number Dom[x] to determine if the transaction is allowed. A UnitID can only be assigned to one protection domain. If a UnitID is assigned to more than one protection domain the results are undefined.

If the request doesn't match on any map register then the address of the request is checked against the DEV table of protection domain 0 to determine if the transaction is allowed.

| Bits | Description |
|---|---|
| 31:29 | Reserved. |
| 28:26 | **Dom1: protection domain 1**. Read-write. This is the protection domain number assigned to Unit1. |
| 25:23 | Reserved. |
| 22:20 | **Dom0: protection domain 0**. Read-write. This is the protection domain number assigned to Unit0. |
| 19:12 | **BusNu: bus number**. Read-write. |
| 11 | **Valid1: UnitID 1 valid**. Read-write. 1=Enable DEV checking for Unit1 and Dom1. |
| 10:6 | **Unit1: internal UnitID 1**. Read-write. |
| 5 | **Valid0: UnitID 0 valid**. Read-write. 1=Enable DEV checking for Unit0 and Dom0. |
| 4:0 | **Unit0: internal UnitID 0**. Read-write. |

### D18F3xF8_x3 DEV Capabilities Register

| Bits | Description |
|---|---|
| 31:24 | Reserved. |
| 23:16 | **NMaps: number of map registers implemented**. Read-only. Reset: 04h. Specifies the number of instantiations of D18F3xF8_x2 of the UID format. |
| 15:8 | **NDomains: number of protection domains implemented**. Read-only. Reset: 08h. Specifies the number of protection domains and the number of instantiations of D18F3xF8_x0 and D18F3xF8_x1. |
| 7:0 | **Revision: DEV register-set revision number**. Read-only. Reset: 02h. Indicates support for D18F3xF8_x4[SecureGfxMode]. |

### D18F3xF8_x4 DEV Control Register

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:9 | Reserved. |

| | |
|---|---|
| 8 | **SecureGfxMode: secure graphics mode**. Read-write; set-by-hardware. This bit is set when an SKINIT instruction is executed.1=All access to memory except for accesses to the frame buffer from the GPU are check by the DEV if the DEV is enabled. |
| 7 | Reserved. |
| 6 | **DevTblWalkPrbDis: DEV table walk probe disable**. Read-write. 1=Disable probing of CPU caches during DEV table walks. This bit may be set to improve DEV cache table walk performance when the DEV is in non-cacheable or write-through memory. |
| 5 | **SlDev: secure loader DEV protection enable**. Read; write-0-only; set-by-hardware. This bit is set when an SKINIT instruction is executed. 1=The memory region associated with the SKINIT instruction is protected from DMA access. |
| 4 | **DevInv: invalidate DEV cache**. Read; write-1-only; cleared-when-done. 1=Invalidate the DEV table-walk cache. |
| 3 | **MceEn: MCE reporting enable**. Read-write. 1=Enable reporting of DEV protection violations through a machine check exception. |
| 2 | **IoDis: upstream IO disable**. Read-write; set-by-hardware. This bit is set when an SKINIT instruction is executed. 1=Upstream IO-space accesses are regarded as DEV protection violations. |
| 1 | Reserved. |
| 0 | **DevEn: DEV enable**. Read-write. 1=Enables DMA exclusion vector protection. |

### D18F3xF8_x5 DEV Error Status Register

Cold reset: 0000_0000h.

This register logs DEV protection violations. Bits[7:0], [ErrTypeDest, ErrTypeSrc, ErrTypeAccType], together form the error type field. When a DEV protection violation occurs, then ErrVal is set, the error type is logged, and, if there is an address associated with the transaction, ErrAddrVal is set and the address is recorded in D18F3xF8_x6 and D18F3xF8_x7.

| Bits | Description |
|---|---|
| 31 | **ErrVal: error valid**. Read-write; set-by-hardware. 1=A valid DEV protection violation has been logged in this register. |
| 30 | **ErrOver: error overflow**. Read-write; set-by-hardware. 1=A DEV protection violation was detected while ErrVal was set for a prior violation. DEV protection violations detected while ErrVal is set are not logged in this register. |
| 29 | **ErrAddrVal: error address valid**. Read-write; set-by-hardware. 1=The address saved in D18F3xF8_x6 and D18F3xF8_x7 is the address associated with the error. |
| 28:24 | Reserved. |
| 23:16 | **ModelSpecErr: model specific error**. Read-only. |
| 15:8 | Reserved. |
| 7:5 | **ErrCodeDest: error code destination**. Read-write; set-by-hardware. Specifies the destination of the transaction that resulted in the protection violation.<br>000b = Generic (or could not be determined)<br>001b = DRAM<br>010b = MMIO<br>011b = reserved<br>100b = IO space<br>101b = Configuration<br>110b = reserved<br>111b = reserved |

| | |
|---|---|
| 4:2 | **ErrCodeSrc: error code source**. Read-write; set-by-hardware. Specifies the source of the transaction that resulted in the protection violation.<br>000b = Generic (or could not be determined) 010b = IO device<br>001b = CPU 011b - 111b = reserved |
| 1:0 | **ErrCodeAccType: error code access type**. Read-write; set-by-hardware. Specifies the access type of the transaction that resulted in the protection violation.<br>00b = Generic (or could not be determined) 10b = Write<br>01b = Read 11b = Read-modify-write |

### D18F3xF8_x6 DEV Error Address Low Register

Cold reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:3 | **ErrAddr: error address bits[31:3]**. Read-write; set-by-hardware. See D18F3xF8_x5. |
| 2:0 | Reserved. |

### D18F3xF8_x7 DEV Error Address High Register

Cold reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:8 | Reserved. |
| 7:0 | **ErrAddr: error address bits[39:32]**. Read-write; set-by-hardware. See D18F3xF8_x5. |

### D18F3xFC CPUID Family/Model Register

These values are identical to the values read out through CPUID Fn0000_0001_EAX.

| Bits | Description |
|---|---|
| 31:28 | Reserved. |
| 27:20 | **ExtFamily: extended family**. Read-only. Reset: 3h. |
| 19:16 | **ExtModel: extended model**. Read-only. Reset: X. |
| 15:12 | Reserved. |
| 11:8 | **BaseFamily**. Read-only. Reset: Fh. |
| 7:4 | **BaseModel**. Read-only. Reset: X. |
| 3:0 | **Stepping**. Read-only. Reset: X. |

### D18F3x128 Clock Power/Timing Control 3

| Bits | Description |
|---|---|
| 31:16 | Reserved |
| 15 | **NbPsiVidEn: NB PSI_L enable**. Read-write. Reset: 0. BIOS: See 2.5.1.4.1 [PSI_L Bit]. This bit specifies how the PSI_L bit for VDDNB is controlled. 1=See D18F3x128[NbPsiVid]. 0=The PSI_L bit is always 1. |

| | |
|---|---|
| 14:8 | **NbPsiVid**. Read-write. Reset: 0. If D18F3x128[NbPsiVidEn]==1, this field specifies a VID code that determines the state of the PSI_L bit for the VDDNB plane. Whenever the VID code generated by the processor for the VDDNB plane is less than (voltage is greater than) NbPsiVid, the PSI_L bit is sent as a 1. Whenever the VID code generated by the processor for the VDDNB plane is greater than or equal to (voltage less than or equal to) NbPsiVid, the PSI_L bit is sent as a 0. See 2.5.1.4.1 [PSI_L Bit]. |
| 7 | Reserved. |
| 6:0 | **C6Vid**. Read-write. Reset: Product-specific. Specifies the VID driven in the package PC6 state. See 2.5.3.2.3 [C-state Actions]. This field must be programmed within the limits specified by MSRC001_0071[MaxVid, MinVid]. |

### D18F3x138 Local Hardware Thermal Control (LHTC) Register

See 2.10.3.2 [Local Hardware Thermal Control (LHTC)] for information on LHTC. D18F3x138 is reserved if [The Northbridge Capabilities Register] D18F3xE8[LHTC capable]=0.

| Bits | Description |
|---|---|
| 31 | **LHtcLock: HTC lock**. Read; write-1-only. Reset: 0. SBIOS: 1. 1=LHtcPstateLimit, LHtcHystLmt, LHtcTmpLmt, and LHtcEn are read-only. 0=LHtcPstateLimit, LHtcHystLmt, LHtcTmpLmt, and LHtcEn are read-write. |
| 30:28 | **LHtcPstateLimit: local HTC P-state limit select**. Read-write. Reset: Product-specific. Specifies the P-state limit of all cores when in the LHTC-active state. The LHtcPstateLimit to apply is not changed if the value of this field is greater than MSRC001_0061[PstateMaxVal]. LHtcPstateLimit must be greater than or equal to its reset value (same or lower performing P-state). See 2.10.3.2 [Local Hardware Thermal Control (LHTC)]. |
| 27:24 | **LHtcHystLmt: local HTC hysteresis**. Read-write. Reset: Product-specific. The processor exits the LHTC-active state when Tctl is less than LHTC temperature limit (LHtcTmpLmt) minus LHTC hysteresis (LHtcHystLmt).<br>Bits: Fh-0h — Description: <LHtcHystLmt*0.5> |
| 23 | **LHtcSlewSel: local HTC slew-controlled temperature select**. Read-write. Reset: 0. 1=LHTC logic is driven by the slew-controlled temperature, Tctl, specified in [The Reported Temperature Control Register] D18F3xA4. 0=LHTC logic is driven by the measured control temperature of the core with no slew controls. |
| 22:16 | **LHtcTmpLmt: local HTC temperature limit**. Read-write. Reset: Product-specific. The processor enters the LHTC-active state when Tctl reaches or exceeds the temperature limit defined by this register. LHtcTmpLmt must be the same or lower than its reset value.<br>Bits: 7Fh-00h — Description: <LHtcTmpLmt*0.5 + 52> |
| 15:13 | Reserved. |
| 12 | **LHtcActSts: local HTC-active status**. Read; Write-1-to-clear. Reset: 0. This bit is set by hardware when the processor enters the local HTC-active state. It is cleared by writing a 1 to it. |
| 11:9 | Reserved. |
| 8 | **LHtcAct: local HTC-active state**. Read-only. Reset: 0. 1=The processor is currently in the local HTC-active state. 0=The processor is not in the local HTC-active state. |

| | |
|---|---|
| 7:1 | Reserved. |
| 0 | **LHtcEn: local HTC enable**. Read-write. Reset: 0. BIOS: IF (D18F3x138[LHtcTmpLmt]==0) THEN 0 ELSE 1 ENDIF. 1=LHTC is enabled; the processor is capable of entering the local HTC-active state. |

### D18F3x15C DPM Voltage Control Register

| Bits | Description |
|---|---|
| 31 | Reserved. |
| 30:24 | **SclkVidLevel3**. Read-write. Reset: Product-specific. See D18F3x15C[SclkVidLevel0]. |
| 23 | Reserved. |
| 22:16 | **SclkVidLevel2**. Read-write. Reset: Product-specific. See D18F3x15C[SclkVidLevel0]. |
| 15 | Reserved. |
| 14:8 | **SclkVidLevel1**. Read-write. Reset: Product-specific. See D18F3x15C[SclkVidLevel0]. |
| 7 | Reserved. |
| 6:0 | **SclkVidLevel0**. Read-write. Reset: Product-specific. Specifies a voltage level used for various NB power management features. See 2.5.5 [Root Complex Power Management]. See the AMD Voltage Regulator Specification, #40182 for encodings. If the VID code specified is 00h, software should consider the VID code invalid. |

### D18F3x17C In-Flight Queue Extended Buffer Allocation Register

Read-write. BIOS: 2.9.3.5.

| Bits | Description |
|---|---|
| 31:14 | Reserved. |
| 13:8 | **HiPriNPBC: high priority channel non-posted buffer count**. Reset: 0. |
| 7:6 | Reserved. |
| 5:0 | **HiPriPBC: high priority channel posted buffer count**. Reset: 0h. |

### D18F3x180 Extended NB MCA Configuration Register

Reset: 0000_0000h. This register is an extension of [The MCA NB Configuration Register] D18F3x44.

| Bits | Description |
|---|---|
| 31:22 | Reserved. |
| 21 | **SyncFloodOnCpuLeakErr: sync flood on CPU leak error enable**. Read-write. BIOS: 1. 1=A sync flood is generated when one of the cores encounters an uncorrectable error which cannot be contained to the process on the core. |
| 20:8 | Reserved. |
| 7 | **SyncFloodOnTgtAbtErr**. Read-write. 1=Enable sync flood on generated or received responses that indicate target aborts. |
| 6 | Reserved. |

| | |
|---|---|
| 5 | **DisPciCfgCpuMstAbtRsp**. Read-write. BIOS: 1b. 1=Disable MCA error reporting for master abort responses to CPU-initiated configuration accesses. It is recommended that this bit be set in order to avoid MCA exceptions being generated from master aborts for PCI configuration accesses, which are common during device enumeration. |
| 4 | **MstAbtChgToNoErrs**. Read-write. 1=Signal no errors instead of master abort in link response packets to IO devices on detection of a master abort condition. When MstAbtChgToNoErrs and D18F3x44[IoMstAbortDis] are both set, MstAbtChgToNoErrs takes precedence. |
| 3 | **DatErrChgToTgtAbt**. Read-write. 1=Signal target abort instead of data error in link response packets to IO devices (for Gen1 link compatibility). |
| 2 | **WDTCntSel[3]: watchdog timer count select bit[3]**. Read-write. BIOS: 0. See D18F3x44[WDTCntSel]. |
| 1:0 | Reserved. |

### D18F3x188 NB Extended Configuration Register

| Bits | Description |
|---|---|
| 31:28 | **FeArbCpuWeightOverHiPrio: NB front-end round-robin arbiter CPU weight over high priority channel**. Read-write. Reset: 4h. BIOS: 1h. This value is the number of times (out of 16) that the arbiter favors the CPU when both a CPU and the High Priority channel are requesting.<br>• FeArbCpuWeightOverHiPrio must be >= 1.<br>When two CPUs are enabled each CPU is granted half of this value on average. |
| 27:24 | **FeArbCpuWeightOverLoPrio: NB front-end round-robin arbiter CPU weight over low priority channel**. Read-write. Reset: Bh. BIOS: Bh. This value is the number of times (out of 16) that the arbiter favors the CPU when both a CPU and the Low Priority channel are requesting.<br>• FeArbCpuWeightOverLoPrio must be >= FeArbCpuWeightOverHiPrio.<br>When two CPUs are enabled each CPU is granted half of this value on average. |
| 23 | **EnCpuSerRdBehindIoRd: enable CPU serialization of reads behind IO reads**. Read-write. Reset: 0. BIOS: 0. 1=From the same CPU, sized reads are serialized behind IO reads. 0=From the same CPU, sized reads are not serialized behind IO reads. |
| 22 | **EnCpuSerRdBehindNpIoWr: enable CPU serialization of reads behind non-posted IO writes**. Read-write. Reset: 0. BIOS: See 2.9.3.5. 1=From the same CPU, sized reads are serialized behind non-posted IO writes. 0=From the same CPU, sized reads are not serialized behind non-posted IO writes. |
| 21 | **EnCpuSerWrBehindIoRd: enable CPU serialization of writes behind IO reads**. Read-write. Reset: 0. BIOS: 0. 1=From the same CPU, sized writes are serialized behind IO reads. 0=From the same CPU, sized writes are not serialized behind IO reads. |
| 20:0 | Reserved. |

### D18F3x1CC IBS Control Register

Reset: 0000_0000h. This register can also be read from MSRC001_103A.

| Bits | Description |
|---|---|
| 31:9 | Reserved. |
| 8 | **LvtOffsetVal: local vector table offset valid**. Read-write. BIOS: 1. 1=The offset in LvtOffset is valid. |

| | |
|---|---|
| 7:4 | Reserved. |
| 3:0 | **LvtOffset: local vector table offset**. Read-write. BIOS: 0. This specifies the address of the IBS LVT entry in the APIC registers as follows: LVT address = (LvtOff shifted left 4 bits) + 500h (see APIC[530:500]). |

### D18F3x1E4 SBI Control Register

See 2.10.2 [Sideband Temperature Sensor Interface (SB-TSI)].

| Bits | Description |
|---|---|
| 31 | **SbiRegWrDn: SBI register write done**. Read-only. Reset: 1b. 1=Write to the SBI registers through D18F3x1EC has completed. 0=Write to the SBI registers in progress. |
| 30:7 | Reserved. |
| 6:4 | **SbiAddr: SMBus-based sideband interface address**. Read-write. Cold reset: specified by the SA[2:0] strap pins (value matches the pins until the deassertion of RESET_L for a cold reset only; value is not changed by a warm reset); 000b in products that do not include SA[2:0] pins. Specifies bits[3:1] of the SMBus address of the processor SBI ports. SMBus address bits [3:1] = {~SA[2],SA[1:0]}. |
| 3:2 | Reserved. |
| 1 | **SbTsiDis: SMBus-based sideband interface thermal sensor disable**. Read-only. Reset: Product-specific. 1=The processor does not support SMBus-based SBI thermal sensor protocol. |
| 0 | Reserved. |

### D18F3x1E8 SBI Address Register

Reset: 0000_0000h.

The SB-TSI registers can be directly accessed by the processor using D18F3x1E8 and D18F3x1EC. Access to these registers is accomplished as follows:

- Reads:
  - Write the register number to D18F3x1E8[SbiRegAddr].
  - Read the register contents from D18F3x1EC.

- Writes:
  - Write the register number to D18F3x1E8[SbiRegAddr].
  - Write all 32 bits to the register data to D18F3x1EC.

| Bits | Description |
|---|---|
| 31:8 | Reserved. |
| 7:0 | **SbiRegAddr: SBI SMBus register address**. Read-write. This field specifies the 8-bit address of the SB-TSI register to access. |

#### D18F3x1EC SBI Data Register

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:8 | Reserved. |
| 7:0 | **SbiRegDat0: SBI SMBus register data**. Read-write. This field specifies the data to be read or written to the SBI register selected by D18F3x1E8. |

#### D18F3x1F0 Product Information Register

| Bits | Description |
|---|---|
| 31:16 | Reserved. |
| 15:0 | **BrandId**. Read-only. Reset: Product-specific. Brand identifier. This is identical to CPUID Fn8000_0001_EBX[BrandId]. |

### 3.11 Device 18h Function 4 Configuration Registers

See 3.1 [Register Descriptions and Mnemonics] for a description of the register naming convention. See 2.7 [Configuration Space] for details about how to access this space.

#### D18F4x00 Device/Vendor ID Register

Reset: 1704_1022h.

| Bits | Description |
|---|---|
| 31:16 | **DeviceID: device ID**. Read-only. |
| 15:0 | **VendorID: vendor ID**. Read-only. |

#### D18F4x04 Status/Command Register

Read-only. Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:16 | **Status**. |
| 15:0 | **Command**. |

### D18F4x08 Class Code/Revision ID Register

Reset: 0600_0000h.

| Bits | Description |
|---|---|
| 31:8 | **ClassCode**. Read-only. Provides the host bridge class code as defined in the PCI specification. |
| 7:0 | **RevID: revision ID**. Read-only. |

### D18F4x0C Header Type Register

Reset: 0080_0000h.

| Bits | Description |
|---|---|
| 31:0 | **HeaderTypeReg**. Read-only. These bits are fixed at their default values. The header type field indicates that there are multiple functions present in this device. |

### D18F4x34 Capabilities Pointer Register

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:8 | Reserved. |
| 7:0 | **CapPtr: capabilities pointer**. Read-only. |

### D18F4x104 TDP Lock Accumulator

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:29 | Reserved. |
| 28:17 | **TdpLockDivRateCpu: per CPU TDP lock divisor rate**. Read-write. BIOS: 190h. Specifies the rate at which the per-core CPB power consumption approximation is changed as specified in TdpLockDivValCpu. See 2.5.3.1.1 [Core Performance Boost (CPB)].<br>Bits: Definition<br>FFFh-000h: <TdpLockDivRateCpu * 5.12us> |
| 16:15 | **TdpLockDivValCpu: per CPU TDP lock divisor value**. Read-write. BIOS: 1. See D18F4x104[TdpLockDivRateCpu].<br>Bits: Definition<br>00b: Divide by 1.<br>01b: Divide by 2.<br>10b: Divide by 4.<br>11b: Clear the per-core CPB power consumption approximation. |
| 14 | Reserved. |

| | |
|---|---|
| 13:2 | **TdpLockDivRate: TDP lock divisor rate**. Read-write. BIOS: 190h. Specifies the rate at which the all-core ‘s CPB power consumption approximation is updated as specified in TdpLockDivVal. See 2.5.3.1.1 [Core Performance Boost (CPB)].<br>Bits: Definition<br>FFFh-000h: <TdpLockDivRate * 5.12us> |
| 1:0 | **TdpLockDivVal: TDP lock divisor value**. Read-write. BIOS: 1. See D18F4x104[TdpLockDivRate].<br>Bits: Definition<br>00b: Divide by 1.<br>01b: Divide by 2.<br>10b: Divide by 4.<br>11b: Clear the all-core ‘s CPB power consumption approximation. |

### D18F4x118 C-state Control 1

Reset: 0000_0000h. BIOS: 2.5.3.2.9.

D18F4x118 and D18F4x11C consist of eight identical 8-bit registers, one for each C-state Action Field (CAF) associated with an IO address that is read to enter C-states. See 2.5.3.2 [C-states].

| Bits | Description |
|---|---|
| 31:27 | Reserved. |
| 26:24 | **CstAct3: C-state action field 3**. Read-write. Specifies the actions attempted by the core when software reads from the IO address specified by MSRC001_0073[CstateAddr + 3]. See Table 122. |
| 23:19 | Reserved. |
| 18:16 | **CstAct2: C-state action field 2**. Read-write. Specifies the actions attempted by the core when software reads from the IO address specified by MSRC001_0073[CstateAddr + 2]. See Table 122. |
| 15:11 | Reserved. |
| 10:8 | **CstAct1: C-state action field 1**. Read-write. Specifies the actions attempted by the core when software reads from the IO address specified by MSRC001_0073[CstateAddr + 1]. See Table 122. |
| 7:3 | Reserved. |
| 2:0 | **CstAct0: C-state action field 0**. Read-write. Specifies the actions attempted by the core when software reads from the IO address specified by MSRC001_0073[CstateAddr]. See Table 122. |

**Table 122: C-state action field definition**

| Bits | Description |
|---|---|
| 7:1 | Reserved. |
| 0 | **C6Enable**. Specifies whether the core attempts to enter CC6 and the package attempts to enter PC6. 1=Attempt to enter CC6 and PC6. 0=Do not attempt to enter CC6 and PC6. See 2.5.3.2.3 [C-state Actions]. |

### D18F4x11C C-state Control 2

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:27 | Reserved. |
| 26:24 | **CstAct7: C-state action field 7**. Read-write. Specifies the actions attempted by the core when software reads from the IO address specified by MSRC001_0073[CstateAddr + 7]. See Table 122. |
| 23:19 | Reserved. |
| 18:16 | **CstAct6: C-state action field 6**. Read-write. Specifies the actions attempted by the core when software reads from the IO address specified by MSRC001_0073[CstateAddr + 6]. See Table 122. |
| 15:11 | Reserved. |
| 10:8 | **CstAct5: C-state action field 5**. Read-write. Specifies the actions attempted by the core when software reads from the IO address specified by MSRC001_0073[CstateAddr + 5]. See Table 122. |
| 7:3 | Reserved. |
| 2:0 | **CstAct4: C-state action field 4**. Read-write. Specifies the actions attempted by the core when software reads from the IO address specified by MSRC001_0073[CstateAddr + 4]. See Table 122. |

### D18F4x120 C-state Policy Control 1

Reset: 1000_0000h.

| Bits | Description |
|---|---|
| 31 | **CstateMsgDis: C-state messaging disable**. Read-write. BIOS: 1. Specifies whether a message is sent to the FCH when a package C-state transition occurs. 0=Send message. 1=Do not send message. See 2.5.3.2.4.2 [FCH Messaging] for details. |
| 30:29 | Reserved. |
| 28:27 | **CoreOffWriPri: CC6 write/read priority**. Read-write. Specifies the priority given to DRAM writes and reads when saving and restoring data to enter or exit the CC6 state.<br>Bits Definition Bits Definition<br>00b Medium. 10b High.<br>01b Low. 11b Variable.<br>See D18F2x118 for more details about DRAM transaction priorities. See 2.5.3.2.3.2 [Core C6 (CC6) State]. |
| 26:25 | Reserved. |
| 24 | **DeepCstAllowMsgEn: deep C-state allow message enable**. Read-write. Specifies whether the processor prevents or allows access to PC6 based on FCH messaging. 0=Access to PC6 does not rely on FCH messaging. 1=Access to PC6 relies on FCH messaging. See 2.5.3.2.4.2 [FCH Messaging]. |
| 23 | Reserved. |
| 22:20 | **FchTO: FCH timeout**. Read-write. Specifies the time to wait for the response from the FCH after requesting access to a package C-state.<br>Bits Definition Bits Definition<br>000b Reserved. 011b 100 microseconds.<br>001b 40 microseconds. 100b 200 microseconds.<br>010b 80 microseconds. 111b-101b reserved<br>See D18F4x120[DeepCstTOPol] and 2.5.3.2.4.2 [FCH Messaging]. |

| | |
|---|---|
| 19 | **DeepCstTOPol: deep C-state timeout policy**. Read-write. Specifies the action to take if a timeout occurs while waiting for a response from the FCH after requesting access to PC6. 0=Prevent access to PC6. 1=Allow access to PC6. See 2.5.3.2.4.2 [FCH Messaging]. |
| 18 | Reserved. |
| 17:16 | **DeepCstDMATrackEn**. Read-write. Specifies the type of DMA activity to track when determining whether to allow access to PC6. See D18F4x120[CstDMATrackHyst] and 2.5.3.2.4.1 [DMA Tracking] for more details.<br>Bits: Definition<br>00b: DMA tracking disabled.<br>01b: DMA tracking enabled, track coherent traffic only.<br>10b: DMA tracking enabled, track non-coherent traffic only.<br>11b: DMA tracking enabled, track both coherent and non-coherent traffic. |
| 15:13 | **CstDMATrackHyst: deep C-state DMA tracking hysteresis**. Read-write. Specifies the hysteresis time for DMA activity tracking.<br>Bits: Definition<br>000b: Reserved.<br>001b: 10 microseconds.<br>010b: 20 microseconds.<br>011b: 40 microseconds.<br>100b: 80 microseconds.<br>101b: 100 microseconds.<br>11xb: Reserved.<br>See 2.5.3.2.4.1 [DMA Tracking] for more details. |
| 12:0 | Reserved. |

### D18F4x124 C-state Monitor Control 1

Reset: 0000_0000h.

This register controls the various system activity monitors that can be used to limit or allow entry into certain C-states. See 2.5.3.2.4 [C-state Request Monitors] for details.

| Bits | Description |
|---|---|
| 31 | **IntMonWakeEn: interrupt monitor wake enable**. Read-write. Specifies whether a Timer Tick message from the FCH with encoding 1111b will automatically wake all cores from a C-state. 0=Do not wake cores. 1=Wake cores. See2.5.3.2.4.3 [Interrupt Monitors]. |
| 30:27 | **IntMonIntrvl: interrupt monitor interval**. Read-only. Specifies the time between timer tick interrupts last reported by the FCH. See2.5.3.2.4.3 [Interrupt Monitors].<br>Bits: Definition<br>0000b: No interval reported.<br>0001b: 0ms to 1ms.<br>0010b: 1ms to 2ms.<br>0011b: 2ms to 3ms.<br>0100b: 3ms to 4ms.<br>0101b: 4ms to 5ms.<br>0110b: 5ms to 6ms.<br>0111b: 6ms to 7ms.<br>1000b: 7ms to 8ms.<br>1001b: 8ms to 9ms.<br>1010b: 9ms to 10ms.<br>1011b: 10ms to 15ms.<br>1100b: >= 15ms<br>1101b: reserved.<br>1110b: reserved.<br>1111b: reserved. |

<table>
<tr><td>26:23</td><td>IntMonPkgC6Lmt: interrupt monitor package C6 limit. Read-write. BIOS: 1010b. Specifies the threshold for disallowing access to the package C6 state. See 2.5.3.2.4.3 [Interrupt Monitors].<br>IF (D18F4x124[TimerTickIntvlScale] == 1) THEN<br><u>Bits</u> <u>Threshold</u><br>0000b PC6 entry allowed<br>1001b-0001b IntMonPkgC6Lmt * 50us<br>1110b-1010b ((5 * IntMonPkgC6Lmt) - 40) * 50us<br>1111b 2000us<br>ELSE<br><u>Bits</u> <u>Threshold</u><br>0000b PC6 entry allowed<br>1001b-0001b IntMonPkgC6Lmt * 1ms<br>1110b-1010b ((5 * IntMonPkgC6Lmt) - 40) * 1ms<br>1111b 40ms<br>ENDIF.</td></tr>
<tr><td>22</td><td>IntMonPkgC6En: interrupt monitor package C6 enable. Read-write. BIOS: 0. Specifies whether the interrupt monitor is enabled for the package C6 state. 0=Disabled. 1=Enabled.</td></tr>
<tr><td>21:18</td><td>IntMonCC6Lmt: interrupt monitor core C6 limit. Read-write. BIOS: 0100b. Specifies the threshold for disallowing access to the core C6 state. See: D18F4x124[IntMonPkgC6Lmt]. See 2.5.3.2.4.3 [Interrupt Monitors].</td></tr>
<tr><td>17</td><td>IntMonCC6En: interrupt monitor core C6 enable. Read-write. BIOS: 1. Specifies whether the interrupt monitor is enabled for the core C6 state. 0=Disabled. 1=Enabled.</td></tr>
<tr><td>16</td><td>TrackTimerTickInterEn: track timer tick interrupt enable. Read-write. BIOS: 1. Specifies the timer tick monitor mode. 0=Interval mode. 1=Duration mode. See 2.5.3.2.4.3 [Interrupt Monitors].</td></tr>
<tr><td>15</td><td>TimerTickIntvlScale: timer tick interval scale. Read-write. BIOS: 1. See D18F4x124[IntMonPkgC6Lmt].</td></tr>
<tr><td>14</td><td>MonitorEnableMode: monitor enable mode. Read-write. Specifies when the C0 residency counter for each core stops counting in relation to core C-state entry. 0=The C0Timers stop counting on any non-CC0 C-state entry. 1=The C0Timers stop counting only when the core enters the CC6 state. See 2.5.3.2.4.4 [Residency Monitors].</td></tr>
<tr><td>13:7</td><td>Reserved.</td></tr>
<tr><td>6:4</td><td>C0MonCC6Cntr: C0 monitor core C6 counter. Read-write. Specifies the threshold for the C0 residency counter. See 2.5.3.2.4.4 [Residency Monitors].</td></tr>
<tr><td>3:1</td><td>C0MonCC6Lmt: C0 monitor core C6 limit. Read-write. Specifies the time threshold for the last C0 residency before incrementing the C0 residency counter. See 2.5.3.2.4.4 [Residency Monitors].<br><u>Bits</u> <u>Residency</u> <u>Bits</u> <u>Residency</u><br>000b Reserved 100b <= 800 us<br>001b <= 100 us 101b <= 1 ms<br>010b <= 200 us 110b <= 2 ms<br>011b <= 400 us 111b >2 ms</td></tr>
<tr><td>0</td><td>C0MonCC6En: C0 monitor core C6 enable. Read-write. Specifies whether the C0 residency monitor is enabled for the CC6 state. 0=Disabled. 1=Enabled. See 2.5.3.2.4.4 [Residency Monitors].</td></tr>
</table>

**D18F4x128 C-state Monitor Control 2**

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:10 | Reserved. |
| 9 | **NonC0MonCoreOffMode**. Read-write. 0=The non-C0 residency monitor relies on the most recent non-C0 residency only. 1=The non-C0 residency monitor uses an internal counter to track multiple non-C0 residencies. See 2.5.3.2.4.4.2 [Non-C0 residency Monitor]. |
| 8:6 | **NonC0MonCoreOffCntr**. Read-write. When D18F4x128[NonC0MonCoreOffMode]==1, this field specifies the limit value for the non-C0 residency monitor internal counter. See 2.5.3.2.4.4.2 [Non-C0 residency Monitor]. |
| 5:1 | **NonC0MonCC6Lmt: non-C0 monitor for core C6 limit**. Read-write. Specifies the non-C0 residency time required on a single non-C0 entry before transitions to CC6 are allowed. See 2.5.3.2.4.4.2 [Non-C0 residency Monitor].<br>IF (D18F4x128[NonC0MonCoreOffMode]==1) THEN<br>Bits: 00h Reserved; 01h <= 200 us; 02h <= 250 us; 03h <= 300 us; 04h <= 350 us; 05h <= 400 us; 06h <= 450 us; 07h <= 500 us; 08h <= 550 us; 09h <= 600 us; 0Ah <= 650 us; 0Bh <= 700 us; 0Ch <= 800 us; 0Dh <= 1 ms; 0Eh <= 1.5 ms; 0Fh <= 2 ms; 1Fh-10h Reserved.<br>ELSE<br>Bits: 00h Reserved; 1Fh-01h <= NonC0Timer0 * 50us<br>ENDIF |
| 0 | **NonC0MonCC6En: non-C0 monitor for core C6 enable**. Read-write. Specifies whether the non-C0 residency monitor is enabled for the CC6 state. 1=Enabled. 0=Disabled. See 2.5.3.2.4.4.2 [Non-C0 residency Monitor]. |

### D18F4x12C C6 Base

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:16 | Reserved. |
| 15:0 | **C6Base[39:24]**. Write-once. BIOS: 2.9.6. Specifies the DRAM base address of the memory region used to store data for the CC6 C-state. See 2.5.3.2.3.2 [Core C6 (CC6) State] and 2.9.6 [DRAM CC6/PC6 Storage]. |

### D18F4x134 C-state Monitor Control 3

Reset: 0000_0000h. BIOS: C5455100h. See 2.5.3.2.4.3 [Interrupt Monitors].

<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>31:27</td><td>IntRateCC6DecrRate: interrupt rate monitor CC6 decrement rate. Read-write. Specifies the rate at which the CC6 interrupt counter is decremented. See D18F4x134[IntRatePkgC6DecrRate] for encodings.</td></tr>
<tr><td>26:24</td><td>IntRateCC6BurstLen: interrupt rate monitor CC6 burst length. Read-write. If the processor receives multiple interrupts within the time window specified by this field, the CC6 interrupt counter is only incremented by one. See D18F4x134[IntRatePkgC6BurstLen] for encodings.</td></tr>
<tr><td>23:20</td><td>IntRateCC6Threshold: interrupt rate monitor CC6 threshold. Read-write. Specifies the threshold the CC6 interrupt counter must reach before access to CC6 is denied.</td></tr>
<tr><td>19:16</td><td>IntRateCC6MaxDepth: interrupt rate monitor CC6 maximum counter depth. Read-write. Specifies the saturation point of the CC6 interrupt counter.</td></tr>
<tr><td>15:11</td><td>IntRatePkgC6DecrRate: interrupt rate monitor PC6 decrement rate. Read-write. Specifies the rate at which the PC6 interrupt counter is decremented.
<table>
<tr><th>Bits</th><th>Decrement Rate</th></tr>
<tr><td>00h</td><td>Reserved.</td></tr>
<tr><td>1Fh-01h</td><td>IntRatePkgC6DecrRate * 50us</td></tr>
</table></td></tr>
<tr><td>10:8</td><td>IntRatePkgC6BurstLen: interrupt rate monitor PC6 burst length. Read-write. If the processor receives multiple interrupts within the time window specified by this field, the PC6 interrupt counter is only incremented by one.
<table>
<tr><th>Bits</th><th>Burst Length</th><th>Bits</th><th>Burst Length</th></tr>
<tr><td>000b</td><td>80 ns</td><td>100b</td><td>75 us</td></tr>
<tr><td>001b</td><td>10 us</td><td>101b</td><td>100 us</td></tr>
<tr><td>010b</td><td>20 us</td><td>110b</td><td>125 us</td></tr>
<tr><td>011b</td><td>50 us</td><td>111b</td><td>150 us</td></tr>
</table></td></tr>
<tr><td>7:4</td><td>IntRatePkgC6Threshold: interrupt rate monitor PC6 threshold. Read-write. Specifies the threshold the PC6 interrupt counter must reach before access to PC6 is denied.</td></tr>
<tr><td>3:0</td><td>IntRatePkgC6MaxDepth: interrupt rate monitor PC6 maximum counter depth. Read-write. Specifies the saturation point of the PC6 interrupt counter.</td></tr>
</table>

### D18F4x138 SMAF Code DID 0

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:29 | Reserved. |
| 28:24 | **Smaf3Did**. Read-write. Specifies the clock divisor used when entering a low power state associated with SMAF3. See D18F4x13C[Smaf0Did]. |
| 23:21 | Reserved. |
| 20:16 | **Smaf2Did**. Read-write. Specifies the clock divisor used when entering a low power state associated with SMAF2. See D18F4x13C[Smaf0Did]. |
| 15:13 | Reserved. |
| 12:8 | **Smaf1Did**. Read-write. Specifies the clock divisor used when entering a low power state associated with SMAF1. See D18F4x13C[Smaf0Did]. |

| | |
|---|---|
| 7:5 | Reserved. |
| 4:0 | **Smaf0Did**. Read-write. Specifies the clock divisor used when entering a low power state associated with SMAF0. See D18F3x80 and D18F3x84.<br><br>Bits Divisor Bits Divisor<br>00h Reserved 09h Reserved<br>01h Reserved 0Ah Reserved<br>02h Reserved 0Bh Reserved<br>03h Reserved 0Ch Reserved<br>04h Reserved 0Dh /512<br>05h Reserved 0Eh Reserved<br>06h Reserved 0Fh Clocks off<br>07h Reserved 1Fh-10h Reserved<br>08h Reserved<br><br>See D18F4x1A8[SingleHaltCpuDid]. |

**D18F4x13C SMAF Code DID 1**

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:29 | Reserved. |
| 28:24 | **Smaf7Did**. Read-write. Specifies the clock divisor used when entering a low power state associated with SMAF7. See D18F4x13C[Smaf0Did]. |
| 23:21 | Reserved. |
| 20:16 | **Smaf6Did**. Read-write. BIOS: 0Fh. Specifies the clock divisor used when entering a low power state associated with SMAF6. See D18F4x13C[Smaf0Did]. |
| 15:13 | Reserved. |
| 12:8 | **Smaf5Did**. Read-write. Specifies the clock divisor used when entering a low power state associated with SMAF5. See D18F4x13C[Smaf0Did]. |
| 7:5 | Reserved. |
| 4:0 | **Smaf4Did**. Read-write. BIOS: 0Fh. Specifies the clock divisor used when entering a low power state associated with SMAF4. See D18F4x13C[Smaf0Did]. |

**D18F4x14C LPMV Scalar 2**

| Bits | Description |
|---|---|
| 31:27 | Reserved. |
| 26 | **CstatePowerSel**. Read-write. Reset: 0. BIOS: 1. Specifies how CPB hardware estimates power for CC1, PC1. 0=Hardware uses the AMD defined value . 1=Hardware uses the AMD defined value multiplied by 4.See 2.5.3.1.1 [Core Performance Boost (CPB)]. |

| | |
|---|---|
| 25:24 | **ApmCstExtPol: CPB C-state exit policy**. Read-write. Reset: 0. BIOS: 01b. Specifies the state CPB uses for power estimation when a core exits a non-C0 C-state. See 2.5.3.1.1 [Core Performance Boost (CPB)].<br>Bits — Power Value<br>00b — C0<br>01b — CC1<br>10b — CC6<br>11b — Reserved. |
| 23:0 | Reserved. |

**D18F4x15C Core Performance Boost Control**

| Bits | Description |
|---|---|
| 31:30 | Reserved. |
| 29 | **BoostEnAllCores: boost enable all cores**. Read-write. Reset: 0. BIOS: 1. Specifies the requirements for a given core to enter a boosted P-state when no other cores are currently in boosted P-states. 1= All cores' power consumption must be below the AMD-defined limit. 0=A given core's power consumption must be below the AMD-defined limit. |
| 28 | **IgnoreBoostThresh: ignore boost threshold**. Read-write. Reset: 0. BIOS: 1. Specifies the requirements for a core to enter a boosted P-state when another core is currently in a boosted P-state. 1=No additional requirements. 0=The given core's power consumption must be below the AMD-defined limit. |
| 27:5 | Reserved. |
| 4:2 | **NumBoostStates: number of boosted states**. Read-only. Reset: Product-specific. Specifies the number of P-states that are considered boosted P-states. See 2.5.3.1.1 [Core Performance Boost (CPB)]. |
| 1:0 | **BoostSrc: boost source**. Read-write. Reset: 0. BIOS: IF ((D18F4x15C[NumBoostStates]==0) \|\| (BatteryPower)) THEN 0. ELSE 1. ENDIF. Specifies whether CPB is enabled or disabled.<br>Bits — Definition<br>00b — Boosting disabled.<br>01b — Boosting enabled.<br>10b — Reserved.<br>11b — Reserved |

**D18F4x164 Fixed Errata**

| Bits | Description |
|---|---|
| 31:0 | **FixedErrata**. Value: Product-specific. See the *Revision Guide for AMD Family 12h Processors* for the definition of this field. |

**D18F4x1A4 C-state Monitor Mask**

Reset: 0000_0000h.

Each bit in each field in this register corresponds to one C-state action field (see D18F4x118 and D18F4x11C). Bit 0 of each field corresponds to D18F4x118[CstAct0], bit 1 corresponds to D18F4x118[CstAct1] and so on.

A monitor is masked for a CAF by setting the corresponding bit to 1. For example, to mask the timer tick monitor for D18F4x11C[CstAct4], set bit 12 in this register to a 1. See 2.5.3.2.4.5 [C-state Monitor Masking] for additional details.

| Bits | Description |
|---|---|
| 31:24 | **C0MonMask: C0 residency monitor mask**. Read-write. BIOS: FFh. Specifies whether the C0 residency monitor is masked. 1=Masked. 0=Unmasked. See 2.5.3.2.4.4 [Residency Monitors]. |
| 23:16 | **NonC0MonMask: non-C0 residency monitor mask**. Read-write. BIOS: FFh. Specifies whether the non-C0 residency monitor is masked. 1=Masked. 0=Unmasked. See 2.5.3.2.4.4 [Residency Monitors]. |
| 15:8 | **TimerTickMonMask: timer tick monitor mask**. Read-write. BIOS: FFh. Specifies whether the timer tick monitor is masked. 1=Masked. 0=Unmasked. See 2.5.3.2.4.3 [Interrupt Monitors]. |
| 7:0 | **IntRateMonMask: interrupt rate monitor mask**. Read-write. BIOS: IF (BatteryPower) THEN FFh. ELSE FCh. ENDIF. Specifies whether the interrupt rate monitor is masked. 1=Masked. 0=Unmasked. See 2.5.3.2.4.3 [Interrupt Monitors]. |

### D18F4x1A8 CPU State Power Management Dynamic Control 0

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:30 | Reserved. |
| 29 | **DramSrHystEnable: dram self-refresh hysteresis enable**. Read-write. BIOS: 1. Specifies whether the DRAM self-refresh hysteresis timer is enabled. 1=Enabled. 0=Disabled. See 2.5.6.1 [DRAM Self-Refresh]. |
| 28:26 | **DramSrHyst: dram self-refresh hysteresis time**. Read-write. BIOS: 101b. Specifies the hysteresis time before DRAM is placed into self-refresh. See 2.5.6.1 [DRAM Self-Refresh].<br>Bits / Delay / Bits / Delay<br>000b Reserved 100b 5 us<br>001b Reserved 101b 10 us<br>010b Reserved 110b 20 us<br>011b 3 us 111b Reserved |
| 25 | **MemTriStateEn: memory clock tri-state enable**. Read-write. BIOS: 1. Specifies whether MEMCLK is tristated while DRAM is in self-refresh. 1=Tristate MEMCLK. 0=Do not tristate MEMCLK. See 2.5.6.1 [DRAM Self-Refresh] for details. |
| 24 | **DramSrEn: dram self-refresh enable**. Read-write. BIOS: See 2.9.3.5. 1=DRAM can be opportunistically placed into self-refresh. 0=DRAM is not placed into self-refresh. This bit should not be set until all cores have been enabled. See D18F0x68[Cpu1En], D18F0x168[Cpu2En, Cpu3En], and 2.5.6.1 [DRAM Self-Refresh]. |
| 23 | **PServiceTmrEn**. Read-write. BIOS: 1. 0=The PService timer is disabled. 1=The PService timer is enabled. See 2.5.3.2.7 [C-state initiated P-state Changes]. |
| 22:20 | **PServiceTmr**. Read-write. BIOS: 001b. Specifies the expiration time of the PService timer. See 2.5.3.2.7 [C-state initiated P-state Changes].<br>Bits / Delay / Bits / Delay<br>000b Reserved 100b 800 us<br>001b 100 us 101b 1 ms<br>010b 200 us 110b 2 ms<br>011b 400 us 111b 5 ms |

<table>
<tr><td>19:17</td><td>PService: service P-state. Read-write. BIOS: 2.5.3.2.9. Specifies the PService state. See 2.5.3.2.7 [C-state initiated P-state Changes]. This field is an index into MSRC001_00[6B:64]. If this field is programmed to 0, the PService state corresponds to the P-state specified by MSRC001_0064. If this field is programmed to 1, the PService state corresponds to the P-state specified by MSRC001_0065, and so on. This field uses hardware P-state numbering. See 2.5.3.1.2.2 [Hardware P-state Numbering] for details.<br>PService must be programmed to lowest-performance P-state displayed to the operating system or to any lower-performance P-state.</td></tr>
<tr><td>16</td><td>Reserved.</td></tr>
<tr><td>15</td><td>CpuProbEn: Cpu probe enable. Read-write. BIOS:0. Specifies the core frequency used to service probes when ramping from CC1 or PC1. See 2.5.3.2.5 [C-states and Probe Requests].</td></tr>
<tr><td>14:10</td><td>Reserved.</td></tr>
<tr><td>9:5</td><td>AllHaltCpuDid. Read-write. BIOS: 0Fh. Specifies the divisor used when entering PC1 with or without auto-Pmin. See 2.5.3.2.3.3 [Package C1 (PC1) State]. This field must be set to a divisor deeper than D18F4x1A8[SingleHaltCpuDid] or undefined behavior may result.<br><br>
<table>
<tr><th>Bits</th><th>Divisor</th><th>Bits</th><th>Divisor</th></tr>
<tr><td>00h</td><td>Reserved</td><td>09h</td><td>Reserved</td></tr>
<tr><td>01h</td><td>Reserved</td><td>0Ah</td><td>Reserved</td></tr>
<tr><td>02h</td><td>Reserved</td><td>0Bh</td><td>Reserved</td></tr>
<tr><td>03h</td><td>Reserved</td><td>0Ch</td><td>/128</td></tr>
<tr><td>04h</td><td>Reserved</td><td>0Dh</td><td>/512</td></tr>
<tr><td>05h</td><td>Reserved</td><td>0Eh</td><td>Reserved</td></tr>
<tr><td>06h</td><td>Reserved</td><td>0Fh</td><td>Clocks off</td></tr>
<tr><td>07h</td><td>Reserved</td><td>1Fh-10h</td><td>Reserved</td></tr>
<tr><td>08h</td><td>Reserved</td><td></td><td></td></tr>
</table>
</td></tr>
<tr><td>4:0</td><td>SingleHaltCpuDid. Read-write. BIOS: 08h. Specifies the divisor used when entering CC1 See 2.5.3.2.3.1 [Core C1 (CC1) State].<br><br>
<table>
<tr><th>Bits</th><th>Divisor</th><th>Bits</th><th>Divisor</th></tr>
<tr><td>00h</td><td>/1</td><td>09h</td><td>Reserved</td></tr>
<tr><td>01h</td><td>/1.5</td><td>0Ah</td><td>Reserved</td></tr>
<tr><td>02h</td><td>/2</td><td>0Bh</td><td>Reserved</td></tr>
<tr><td>03h</td><td>/3</td><td>0Ch</td><td>/128</td></tr>
<tr><td>04h</td><td>/4</td><td>0Dh</td><td>/512</td></tr>
<tr><td>05h</td><td>/6</td><td>0Eh</td><td>Reserved</td></tr>
<tr><td>06h</td><td>/8</td><td>0Fh</td><td>Reserved</td></tr>
<tr><td>07h</td><td>/12</td><td>1Fh-10h</td><td>Reserved</td></tr>
<tr><td>08h</td><td>/16</td><td></td><td></td></tr>
</table>
<br>
• If MSRC001_00[6B:64][CpuDid] of the current P-state is greater than or equal to SingleHaltCpuDid, then no frequency change is made when entering the low-power state associated with this register.<br>
• If D18F3xD4[ShallowHaltDidAllow]==0, only divisors of /16 and deeper may be used or undefined behavior may result.<br>
• The CPU COF == 100MHz * (CpuFid + 10h) / (divisor specified by this field).</td></tr>
</table>

**D18F4x1AC CPU State Power Management Dynamic Control 1**

| Bits | Description |
|---|---|
| 31 | Reserved. |
| 30 | **CstPminEn: C-state Pmin enable**. Read-write. Reset: 0. BIOS: 2.5.3.2.9. Specifies whether auto-Pmin is enabled. 1=Enabled. 0=Disabled. See 2.5.3.2.7 [C-state initiated P-state Changes]. |
| 29 | **CoreC6Dis: core C6 disable**. Read-write. Reset:0. See D18F4x1AC[CoreC6Cap]. |
| 28 | **PkgC6Dis: package C6 disable**. Read-write. Reset:0. See D18F4x1AC[PkgC6Cap]. |
| 27 | **CoreC6Cap: core C6 capable**. Read-only. Reset: Product-specific. Along with D18F4x1AC[CoreC6Dis], this field specifies whether the processor is capable of removing power from a core when in the CC6 state. If either D18F4x1AC[CoreC6Dis]==1 or D18F4x1AC[CoreC6Cap]==0, the processor cannot remove power from cores. If both D18F4x1AC[CoreC6Dis]==0 and D18F4x1AC[CoreC6Cap]==1, the processor can remove power from cores. See 2.5.3.2.3 [C-state Actions]. |
| 26 | **PkgC6Cap: package C6 capable**. Read-only. Reset: Product-specific. Along with D18F4x1AC[PkgC6Dis], this field specifies whether the processor is capable of entering the PC6 state. If either D18F4x1AC[PkgC6Dis]==1 or D18F4x1AC[PkgC6Cap]==0, the processor cannot enter the PC6 state. If both D18F4x1AC[PkgC6Dis]==0 and D18F4x1AC[PkgC6Cap]==1, the processor can enter the PC6 state. See 2.5.3.2.3 [C-state Actions]. |
| 25:19 | Reserved. |
| 18:16 | **PstateIdCoreOffExit**. Read-write. Reset: 0. BIOS: 2.5.3.2.9. When exiting the package C6 state, the core transitions to the P-state specified by this register. See 2.5.3.2.7.2 [Exiting PC6]. If PC6 is enabled (see 2.5.3.2.9 [BIOS Requirements for C-state Initialization]), PstateIdCoreOffExit must be programmed to lowest-performance P-state displayed to the operating system or to any lower-performance P-state. Programming this field to 0 causes the core to transition to the last P-state requested by software when exiting package C6. This field uses hardware P-state numbering. See 2.5.3.1.2.2 [Hardware P-state Numbering]. |
| 15:10 | Reserved. |
| 9:5 | **C6Did: CC6 divisor**. Read-write. Reset: 0. BIOS: 0Fh. Specifies the divisor applied to the core when ramping clocks down for CC6. See 2.5.3.2.3.2 [Core C6 (CC6) State].<br><br>Bits / Divisor: 00h Reserved; 01h Reserved; 02h Reserved; 03h Reserved; 04h Reserved; 05h Reserved; 06h Reserved; 07h Reserved; 08h Reserved; 09h Reserved; 0Ah Reserved; 0Bh Reserved; 0Ch /128; 0Dh /512; 0Eh Reserved; 0Fh Clocks off; 1Fh-10h Reserved<br><br>See D18F4x1A8[SingleHaltCpuDid]. |
| 4:0 | Reserved. |

## 3.12 Device 18h Function 5 Configuration Registers

See 3.1 [Register Descriptions and Mnemonics] for a description of the register naming convention. See 2.7 [Configuration Space] for details about how to access this space.

### D18F5x00 Device/Vendor ID Register

Reset: 1718_1022h.

| Bits | Description |
|---|---|
| 31:16 | **DeviceID: device ID**. Read-only. |
| 15:0 | **VendorID: vendor ID**. Read-only. |

### D18F5x04 Status/Command Register

Read-only. Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:16 | **Status**. |
| 15:0 | **Command**. |

### D18F5x08 Class Code/Revision ID Register

Reset: 0600_0000h.

| Bits | Description |
|---|---|
| 31:8 | **ClassCode**. Read-only. Provides the host bridge class code as defined in the PCI specification. |
| 7:0 | **RevID: revision ID**. Read-only. |

### D18F5x0C Header Type Register

Reset: 0080_0000h.

| Bits | Description |
|---|---|
| 31:0 | **HeaderTypeReg**. Read-only. The header type field indicates that there are multiple functions present in this device. |

### D18F5x34 Capabilities Pointer Register

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:8 | Reserved. |
| 7:0 | **CapPtr: capabilities pointer**. Read-only. |

## 3.13 Device 18h Function 6 Configuration Registers

See 3.1 [Register Descriptions and Mnemonics] for a description of the register naming convention. See 2.7 [Configuration Space] for details about how to access this space.

### D18F6x00 Device/Vendor ID Register

Reset: 1716_1022h.

| Bits | Description |
|---|---|
| 31:16 | **DeviceID: device ID**. Read-only. |
| 15:0 | **VendorID: vendor ID**. Read-only. |

### D18F6x04 Status/Command Register

Read-only. Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:16 | **Status**. |
| 15:0 | **Command**. |

### D18F6x08 Class Code/Revision ID Register

Reset: 0600_0000h.

| Bits | Description |
|---|---|
| 31:8 | **ClassCode**. Read-only. Provides the host bridge class code as defined in the PCI specification. |
| 7:0 | **RevID: revision ID**. Read-only. |

### D18F6x0C Header Type Register

Reset: 0080_0000h.

| Bits | Description |
|---|---|
| 31:0 | **HeaderTypeReg**. Read-only. These bits are fixed at their default values. The header type field indicates that there are multiple functions present in this device. |

### D18F6x34 Capabilities Pointer Register

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:8 | Reserved. |
| 7:0 | **CapPtr: capabilities pointer**. Read-only. |

### D18F6x50 Configuration Register Access Control

Reset: 0000_0006h.

| Bits | Description |
|---|---|
| 31:2 | Reserved. |

| | |
|---|---|
| 1 | **CfgAccAddrMode: configuration access address mode**. Read-write. BIOS: 0. Specifies the register range accessible by the SMU. 1=Access to only bus 0 device 18h, function 6, offsets 54h to 7Fh and 154 to 17Fh is supported. 0=Access to all bus 0 device 18h offsets is supported. |
| 0 | Reserved. |

### D18F6x[1,0]54 Dram Arbitration Control FEQ Collision

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31 | **PpMode: Protection Period Mode**. Read-write. BIOS: 0b. 1=The protection period for this register is based on counting 32-byte data packets of display requests. 0=The protection period for this register is based on counting NCLK cycles. |
| 30:24 | Reserved. |
| 23:16 | **FeqHiPrio: FEQ high priority**. Read-write. BIOS: 08h. Protection period since the display opened a rank/bank before a pending FEQ high priority request to the same rank/bank pair can be considered eligible for arbitration. |
| 15:8 | **FeqMedPrio: FEQ medium priority**. Read-write. BIOS: 10h. Protection period since the display opened a rank/bank before a pending FEQ medium priority request to the same rank/bank pair can be considered eligible for arbitration. |
| 7:0 | **FeqLoPrio: FEQ low priority**. Read-write. BIOS: 20h. Protection period since the display opened a rank/bank before a pending FEQ low priority request to the same rank/bank pair can be considered eligible for arbitration. |

### D18F6x[1,0]58 Dram Arbitration Control Display Collision

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:24 | **DispUrgPrio: display urgent priority**. Read-write. BIOS: 00h. Number of NCLK cycles since FEQ opened a rank/bank before a pending display urgent priority request to the same rank/bank pair can be considered eligible for arbitration. |
| 23:16 | **DispHiPrio: display high priority**. Read-write. BIOS: 10h. Number of NCLK cycles since FEQ opened a rank/bank before a pending display high priority request to the same rank/bank pair can be considered eligible for arbitration. |
| 15:8 | **DispMedPrio: display medium priority**. Read-write. BIOS: 20h. Number of NCLK cycles since FEQ opened a rank/bank before a pending display medium priority request to the same rank/bank pair can be considered eligible for arbitration. |
| 7:0 | **DispLoPrio: display low priority**. Read-write. BIOS: 40h. Number of NCLK cycles since FEQ opened a rank/bank before a pending display low priority request to the same rank/bank pair can be considered eligible for arbitration. |

### D18F6x[1,0]5C Dram Arbitration Control FEQ Write Protect

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31 | **PpMode: Protection Period Mode**. Read-write. BIOS: 0b. 1=The protection period for this register is based on counting 32-byte data packets of display requests. When setD18F6x[1,0]6C[FeqHiPrio, FeqMedPrio and FeqLoPrio] must not equal FFh. 0=The protection period for this register is based on counting NCLK cycles. |
| 30:24 | Reserved. |
| 23:16 | **FeqHiPrio: FEQ high priority**. Read-write. BIOS: 08h. Protection period since display write was arbitrated before a pending FEQ high priority read request is eligible for arbitration. |
| 15:8 | **FeqMedPrio: FEQ medium priority**. Read-write. BIOS: 10h. Protection period since display write was arbitrated before a pending FEQ medium priority read request is eligible for arbitration. |
| 7:0 | **FeqLoPrio: FEQ low priority**. Read-write. BIOS: 20h. Protection period since display write was arbitrated before a pending FEQ low priority read request is eligible for arbitration. |

### D18F6x[1,0]60 Dram Arbitration Control Display Write Protect

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:24 | **DispUrgPrio: display urgent priority**. Read-write. BIOS: 00h. Number of NCLK cycles since FEQ write was arbitrated before a pending display urgent priority read request is eligible for arbitration. |
| 23:16 | **DispHiPrio: display high priority**. Read-write. BIOS: 08h. Number of NCLK cycles since FEQ write was arbitrated before a pending display high priority read request is eligible for arbitration. |
| 15:8 | **DispMedPrio: display medium priority**. Read-write. BIOS: 10h. Number of NCLK cycles since FEQ write was arbitrated before a pending display medium priority read request is eligible for arbitration. |
| 7:0 | **DispLoPri: display low priority**. Read-write. BIOS: 20h. Number of NCLK cycles since FEQ write was arbitrated before a pending display low priority read request is eligible for arbitration. |

### D18F6x[1,0]64 Dram Arbitration Control FEQ Read Protect

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31 | **PpMode: Protection Period Mode**. Read-write. BIOS: 0b. 1=The protection period for this register is based on counting 32-byte data packets of display requests. When setD18F6x[1,0]6C[FeqHiPrio, FeqMedPrio and FeqLoPrio] must not equal FFh. 0=The protection period for this register is based on counting NCLK cycles. |
| 30:24 | Reserved. |
| 23:16 | **FeqHiPrio: FEQ high priority**. Read-write. BIOS: 04h. Protection period since display read was arbitrated before a pending FEQ high priority write request is eligible for arbitration. |

| | |
|---|---|
| 15:8 | **FeqMedPrio: FEQ medium priority**. Read-write. BIOS: 08h. Protection period since display read was arbitrated before a pending FEQ medium priority write request is eligible for arbitration. |
| 7:0 | **FeqLoPrio: FEQ low priority**. Read-write. BIOS: 10h. Protection period since display read was arbitrated before a pending FEQ low priority write request is eligible for arbitration. |

**D18F6x[1,0]68 Dram Arbitration Control Display Read Protect**

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:24 | **DispUrgPrio: display urgent priority**. Read-write. BIOS: 00h. Number of NCLK cycles since FEQ read was arbitrated before a pending display urgent priority write request is eligible for arbitration. |
| 23:16 | **DispHiPrio: display high priority**. Read-write. BIOS: 04h. Number of NCLK cycles since FEQ read was arbitrated before a pending display high priority write request is eligible for arbitration. |
| 15:8 | **DispMedPrio: display medium priority**. Read-write. BIOS: 08h. Number of NCLK cycles since FEQ read was arbitrated before a pending display medium priority write request is eligible for arbitration. |
| 7:0 | **DispLoPrio: display low priority**. Read-write. BIOS: 10h. Number of NCLK cycles since FEQ read was arbitrated before a pending display low priority write request is eligible for arbitration. |

**D18F6x[1,0]6C Dram Arbitration Control FEQ Fairness Timer**

Reset: 00FF_FFFFh.

| Bits | Description |
|---|---|
| 31:24 | Reserved. |
| 23:16 | **FeqHiPrio: FEQ high priority**. Read-write. BIOS: 20h. This field defines the number of NCLK cycles a high priority FEQ request must wait before its priority gets elevated to be arbitrated immediately. |
| 15:8 | **FeqMedPrio: FEQ medium priority**. Read-write. BIOS: 40h. This field defines the number of NCLK cycles a medium priority FEQ request must wait before its priority gets elevated to be arbitrated immediately. |
| 7:0 | **FeqLoPrio: FEQ low priority**. Read-write. BIOS: 80h. This field defines the number of NCLK cycles a low priority FEQ request must wait before its priority gets elevated to be arbitrated immediately. |

**D18F6x[1,0]70 Dram Arbitration Control Display Fairness Timer**

Reset: FFFF_FFFFh.

| Bits | Description |
|---|---|
| 31:24 | **DispUrPrio: display urgent priority**. Read-write. BIOS:00h. This field defines the number of NCLK cycles an urgent priority display request must wait before its priority gets elevated to be arbitrated immediately. |
| 23:16 | **DispHiPrio: display high priority**. Read-write. BIOS: 20h. This field defines the number of NCLK cycles a high priority display request must wait before its priority gets elevated to be arbitrated immediately. |

| | |
|---|---|
| 15:8 | **DispMedPrio: display medium priority**. Read-write. BIOS: 40h. This field defines the number of NCLK cycles a medium priority display request must wait before its priority gets elevated to be arbitrated immediately. |
| 7:0 | **DispLoPrio: display low priority**. Read-write. BIOS: 80h. This field defines the number of NCLK cycles a low priority display request must wait before its priority gets elevated to be arbitrated immediately. |

### D18F6x[1,0]74 Dram Idle Page Close Limit

Reset: 0000_001Eh.

| Bits | Description |
|---|---|
| 31:5 | Reserved. |
| 4:0 | **IdleLimit: idle limit**. Read-write. BIOS: 1Eh. If D18F2x[1,0]90[DynPageCloseEn]=0 then this field multiplied by 4 defines the number of NB clock cycles a page will be kept open after last page hit.<br>Bits — Definition<br>1Eh-00h — <4*IdleLimit> NCLK delay.<br>1Fh — Reserved. |

### D18F6x78 Dram Prioritization and Arbitration Control

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:16 | Reserved. |
| 15:8 | **DbeCmdThrottle: Dram controller back-end command throttle**. Read-write. BIOS: See 2.9.3.5. This field defines a limit for 64-byte read or write requests pending in the DRAM controller back-end. If the number of pending requests reaches or exceeds this limit, the DRAM controller front-end applies command throttling.<br>Bits — Limit<br>00h — throttling is disabled<br>FFh-01h — <DbeCmdThrottle> request(s) |
| 7 | Reserved. |
| 6 | **GlcEosDetDis: display end of stream detection disable**. Read-write. BIOS: 0. 0=Display end of stream detection is enabled. 1=Display end of stream detection is disabled. |
| 5:4 | **GlcEosDet: display end of stream detection**. Read-write. BIOS: 11b. This register defines the number of NCLK cycles that the display queue must be empty before a n end of stream event is detected. Upon a display end of stream event the read/write protection counter is reset if it was started by a display request.<br>Bits — Limit<br>00b — <2^GlcEosDet> NCLK cycle(s) |
| 3 | **DispArbCtrl: display arbitration control**. Read-write. BIOS: 0. 1=Display requests, which are delayed by a bank-state-machine-full condition, block lower-priority front-end queue requests from being arbitrated. 0=Display requests, which are delayed by a bank-state-machine-full condition, do not block lower-priority front-end queue requests from being arbitrated. |

| | |
|---|---|
| 2 | **FeqDbePrioEn: FEQ DBE priority enable**. Read-write. BIOS: 1. 1=The DRAM controller back-end priority condition is asserted for high priority front-end queue requests. 0=The DRAM controller back-end priority condition is never asserted for front-end queue requests. |
| 1:0 | **DispDbePrioEn: display DBE priority enable**. Read-write. BIOS: 11b. This field defines which type of display requests assert the priority condition in the DRAM controller back-end.<br>Bits — Definition<br>00b — No display request<br>01b — Display refresh request of any display priority<br>10b — Any display request of urgent priority<br>11b — Display refresh request of urgent priority |

### D18F6x80 Clocking Control Register 1

| Bits | Description |
|---|---|
| 31:27 | Reserved. |
| 26:24 | **CableSafeDisAux[6:4]**. Read-write. Reset:0. 1=Cable safe mode for the Aux channel is disabled. 0=Cable safe mode for the Aux channel is enabled. |
| 23 | Reserved. |
| 22:20 | **CableSafeDisAux[3:1]**. Read-write. Reset:0. 1=Cable safe mode for the Aux channel is disabled. 0=Cable safe mode for the Aux channel is enabled. |
| 19:0 | Reserved. |

### D18F6x90 NB P-state Config Low

See 2.5.4.1 [NB P-states].

| Bits | Description |
|---|---|
| 31 | **NbPsCap: NB P-state capable**. Read-only. Reset: Product-specific. 1=The processor is capable of making NB P-state transitions. 0=The processor is not capable of making NB P-state transitions. |
| 30 | **NbPsCtrlDis: NB P-state control disable**. IF (D18F6x90[NbPsLock]) THEN Read-only. ELSE Read-write. ENDIF. Reset: 0. Specifies whether hardware is responsible for requesting NB P-state transitions. 0=Hardware dynamically requests NB P-state transitions during run-time without additional input from software. 1=Hardware does not make NB P-state transition requests. Whenever this bit is set to 1, hardware automatically makes a transition to NBP0. |
| 29 | **NbPsForceSel: NB P-state force selection**. Read-write. Reset: 0. Specifies the target NB P-state for any forced NB P-state transition. 1=Target NBP1. 0=Target NBP0. See D18F6x90[NbPsForceReq]. |
| 28 | **NbPsForceReq: NB P-state force request**. Read-write. Reset: 0. If D18F6x98[NbPsTransInFlight]==0 when this bit is set, a forced NB P-state transition to the state specified by D18F6x90[NbPsForceSel] is initiated. This transition occurs regardless of the state of D18F6x90[NbPsCtrlDis]. If D18F6x98[NbPsTransInFlight]==1 when this bit is set to 1, the NB P-state transition request is queued until the transition currently in flight finishes. |
| 27:21 | Reserved. |

| | |
|---|---|
| 20 | **NbPsLock: NB P-state lock**. Read; write-1-only. Reset: Product-specific. This field controls the writability of several other fields relating to NB P-states. See the following fields for details:<br>• D18F6x90[NbPsCtrlDis].<br>• D18F6x90[NbPs1Vid].<br>• D18F6x90[NbPs1NclkDiv]. |
| 19:17 | Reserved. |
| 16 | **NbPs1GnbSlowIgn: NB P-state ignore GPU slow signal**. Read-write. Reset: 0. IF (GpuEnabled) THEN BIOS: 0. ELSE BIOS: 1. ENDIF. 0=The GPU driver specifies a level of GPU activity that can cause an NB P-state transition. 1=GPU activity is not taken into account when determining whether to make an NB P-state transition. |
| 15 | Reserved. |
| 14:8 | **NbPs1Vid**. IF (D18F6x90[NbPsLock]) THEN Read-only. ELSE Read-write. ENDIF. Reset: Product-specific. BIOS: 2.5.4.1.1. Specifies the VID code output by the processor for VDDNB when in NBP1. Writes to this field take effect on the next transition from NBP0 to NBP1. See the AMD Voltage Regulator Specification, #40182 for encodings.<br><br>Writing this field while D18F6x90[NbPsCtrlDis] == 0 may result in undefined behavior. Whenever this field is written, software must wait the RampTime specified by D18F3xD8[VSRampSlamTime] before clearing D18F6x90[NbPsCtrlDis] to 0, changing the value of D18F6x90[NbPs1NclkDiv], or changing the value of D18F6x90[NbPsForceSel]. |
| 7 | Reserved. |
| 6:0 | **NbPs1NclkDiv: NBP1 NCLK divisor**. IF (D18F6x90[NbPsLock]) THEN Read-only. ELSE Read-write. ENDIF. Reset: Product-specific. BIOS: 2.5.4.1.1. Specifies the divisor applied to NCLK when in NBP1.<br><br>• Writes that change the value of this field cause NCLK to transition to the new divisor if the processor is currently in NBP1.<br>• Software may only change the value of this field if either:<br>  • The sequence described in 2.9.3 [DCT/DRAM Initialization and Resume] has not been run, or<br>  • DRAM has been placed into self-refresh. See D18F2x[1,0]90[EnterSelfRef].<br><br>See: D18F3xDC[NbPs0NclkDiv]. |

### D18F6x94 NB P-state Config High

Reset: 0000_0000h. See 2.5.4.1 [NB P-states].

| Bits | Description |
|---|---|
| 31:29 | **NbPs0ResTmrMin: NBP0 minimum residency timer**. Read-write. Upon transitioning from NBP1 to NBP0, transitions back to NBP1 are blocked until the amount of time specified by this register has passed.<br>Bits — Residency<br>000b — 0<br>001b — 120ns<br>010b — 100us<br>011b — 500us<br>100b — 1ms<br>101b — 5ms<br>110b — 10ms<br>111b — 50ms |
| 28:26 | **NbPs1ResTmrMin: NBP1 minimum residency timer**. Read-write. Upon transitioning from NBP0 to NBP1, transitions back to NBP0 are blocked until the amount of time specified by this register has passed. See: D18F6x94[NbPs0ResTmrMin]. |
| 25:23 | **NbPsC0Timer: NB P-state C0 timer**. Read-write. BIOS: 100b. Specifies the time any core must be in C0 before a transition from NBP1 to NBP0 is triggered.<br>Bits — Residency<br>000b — 0<br>001b — 120ns<br>010b — 25us<br>011b — 50us<br>100b — 100us<br>101b — 200us<br>110b — 500us<br>111b — 1ms |
| 22:20 | **NbPsNonC0Timer: NB P-state non-C0 timer**. Read-write. Specifies the time all cores must be in a non-C0 C-state before a transition from NBP0 to NBP1 is triggered. See: D18F6x94[NbPsC0Timer]. |
| 19:5 | Reserved. |
| 4 | **NbPs1NoTransOnDma: NB P-state no transitions on DMA**. Read-write. BIOS: 0. 1=DMA traffic prevents NB P-state transitions. 0=DMA traffic do not affect NB P-state transitions. |
| 3 | **CpuPstateThrEn: CPU P-state threshold enable**. Read-write. BIOS: 1. Specifies whether core P-states are used as a threshold for NB P-state transitions. 1=Core P-states are used as a threshold. 0=Core P-states are not used. |
| 2:0 | **CpuPstateThr: CPU P-state threshold**. Read-write. BIOS: IF (D18F4x15C[NumBoostStates] == 0) THEN 1. ELSE 2. ENDIF. When D18F6x94[CpuPstateThrEn]==1, this field specifies the core P-state number that acts as a threshold for NB P-states. |

**D18F6x98 NB P-state Control and Status**

Reset: 0000_0000h. See 2.5.4.1 [NB P-states].

| Bits | Description |
|---|---|
| 31 | **NbPsDbgEn: NB P-state debug enable**. Read-write. For any registers that change context based on NB P-states, this field specifies what causes a context swap. 0=Register context is selected by the current NB P-state. 1=Register context is selected by D18F6x98[NbPsCsrAccSel] regardless of the current NB P-state. See 2.9.3.4.7 [NB P-states for DCT/Dram Initialization and Training]. |
| 30 | **NbPsCsrAccSel: NB P-state register accessibility select**. Read-write. If D18F6x98[NbPsDbgEn]==1, this field specifies the context of any registers that have context swaps based on NB P-state. 0=Registers access the NBP0 context. 1=Registers access the NBP1 context. See 2.9.3.4.7 [NB P-states for DCT/Dram Initialization and Training]. |
| 29:3 | Reserved. |
| 2 | **NbPs1Act: NB P-state 1 active**. Read-only; set-by-hardware. Specifies the current NB P-state. 0=NB is currently in NBP0. 1=NB is currently in NBP1. |
| 1 | **NbPs1ActSts: NB P-state 1 active status**. Read; write-1-to-clear; set-by-hardware. Specifies whether the NB has ever transitioned to NBP1 since the last time this field was cleared by software. 1=NB transitioned to NBP1. |
| 0 | **NbPsTransInFlight: NB P-state transition in flight**. Read-only. Specifies whether an NB P-state transition is in process. 1=NB P-state transition is occurring. 0=All NB P-state transitions completed. |

**D18F6x9C NCLK Reduction Control**

Reset: 0000_00FFh. See 2.5.4.2 [NB Clock Ramping].

| Bits | Description |
|---|---|
| 31:9 | Reserved. |
| 8 | **NclkRampWithDllRelock**. Read-write. BIOS: 1. Specifies whether NCLK ramps up in parallel or serially with the DDR PHY DLL being relocked when exiting NB clock ramping. 0=The DDR PHY DLL is relocked after NCLK is ramped up. 1=The DDR PHY DLL relock is started at the same NCLK begins ramping up. This bit can only be programmed to 1 if all of the following are true, or undefined behavior results:<br>• D18F2x[1,0]90[DisDllShutdownSR]==0.<br>• (the main PLL frequency specified by D18F3xD4[MainPllOpFreqId]) / (the divisor specified by D18F6x9C[NclkRedDiv]) >= 100MHz. |

| | |
|---|---|
| 7 | **NclkRedSelfRefrAlways: NCLK reduction during self-refresh always**. Read-write. BIOS: 1. Specifies which C-state the package must be in to allow NCLK to ramp down when DRAM enters self-refresh. 0=PC6. 1=PC1 or deeper. See 2.5.4.2 [NB Clock Ramping] for details. |
| 6:0 | **NclkRedDiv: NCLK reduction divisor**. Read-write. BIOS: 78h. Specifies the divisor used for NCLK when NCLK is ramped down while DRAM is in self-refresh. The following divisors may be created:<br>Bits — Divisor<br>1Fh-00h — reserved<br>20h — /8<br>3Fh-21h — reserved<br>40h — /16<br>5Fh-41h — reserved<br>60h — /32<br>77h-61h — reserved<br>78h — /56<br>7Eh-79h — reserved<br>7Fh — /128<br>• NclkRedDiv must be programmed to a divisor greater than (lower frequency than) D18F3xDC[NbPs0NclkDiv] and D18F6x90[NbPs1NclkDiv] or undefined behavior may result.<br>• A divisor of /128 may only be used when D18F3xD4[MainPllOpFreqId] specifies a main PLL frequency of 3.2GHz or greater or undefined behavior may result. |

### D18F6x[B8:B0] Package C-state Residency

Reset: 0000_0000h. Each counter in D18F6x[B8:B0] applies to one package C-state and is enabled as follows:

| Register | Function | Enable bit |
|---|---|---|
| D18F6xB0 | Any non-C0 package state | D18F6xE0[PkgNonC0ResEn] |
| D18F6xB4 | Package C1 with auto-Pmin | D18F6xE0[PkgC1ResEn] |
| D18F6xB8 | Package C6 | D18F6xE0[PkgC6ResEn] |

| Bits | Description |
|---|---|
| 31:0 | **PkgCstateResidency**. Read-write; updated-by-hardware. Specifies the time the package is in the corresponding state. Hardware increments this field once for every 80ns spent in the package C-state while the specified enable bit is set to 1. If the processor is servicing a probe when entering a package C-state, this counter does not begin incrementing until that probe service is complete. Any write by software clears this register to 0. See 2.5.3.2.3 [C-state Actions]. |

### D18F6xE0 Power Management Residency Counter Enable

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:3 | Reserved. |
| 2 | **PkgC6ResEn: package C6 residency counter enable**. Read-write. See D18F6x[B8:B0]. |

| | |
|---|---|
| 1 | **PkgC1ResEn: package C1 with auto-Pmin residency counter enable**. Read-write. See D18F6x[B8:B0]. |
| 0 | **PkgNonC0ResEn: package non-C0 residency counter enable**. Read-write. See D18F6x[B8:B0]. |

## 3.14 Device 18h Function 7 Configuration Registers

See 3.1 [Register Descriptions and Mnemonics] for a description of the register naming convention. See 2.7 [Configuration Space] for details about how to access this space.

### D18F7x00 Device/Vendor ID Register

Reset: 1719_1022h.

| Bits | Description |
|---|---|
| 31:16 | **DeviceID: device ID**. Read-only. |
| 15:0 | **VendorID: vendor ID**. Read-only. |

### D18F7x04 Status/Command Register

Read-only. Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:16 | **Status**. |
| 15:0 | **Command**. |

### D18F7x08 Class Code/Revision ID Register

Reset: 0600_0000h.

| Bits | Description |
|---|---|
| 31:8 | **ClassCode**. Read-only. Provides the host bridge class code as defined in the PCI specification. |
| 7:0 | **RevID: revision ID**. Read-only. |

### D18F7x0C Header Type Register

Reset: 0080_0000h.

| Bits | Description |
|---|---|
| 31:0 | **HeaderTypeReg**. Read-only. The header type field indicates that there are multiple functions present in this device. |

### D18F7x34 Capabilities Pointer

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:8 | Reserved. |
| 7:0 | **CapPtr: capabilities pointer**. Read-only. |

### 3.15 Internal System Management Unit (SMU) Registers

See section 3.1 [Register Descriptions and Mnemonics] for a description of the register naming convention.

To read SMU registers, software performs the following sequence:

1. If reading a 16-bit register, write the address to D0F0x64_x4D[SmuAddr]. If reading a 32-bit register, write the address + 1 to D0F0x64_x4D[SmuAddr]. This must be done for all reads.
2. Clear D0F0x64_x4D[ReqType] to 0.
3. If D0F0x64_x4D[ReqToggle]==0, set it to 1. If D0F0x64_x4D[ReqToggle]==1, clear it to 0.
4. Read data from D0F0x64_x4E[SmuReadData].

To write data to a 16-bit SMU register, software performs the following sequence:

1. Write the address to D0F0x64_x4D[SmuAddr].
2. Write the data to D0F0x64_x4D[SmuWriteData].
3. Set D0F0x64_x4D[ReqType] to 1.
4. If D0F0x64_x4D[ReqToggle]==0, set it to 1. If D0F0x64_x4D[ReqToggle]==1, clear it to 0.

To write data to a 32-bit SMU register, software performs the following sequence:

1. Perform steps 1-4 of the 16-bit write process above using the lower 16-bits of data.
2. Program D0F0x64_x4D[SmuAddr]+=1.
3. Write the upper 16-bits of data to D0F0x64_x4D[SmuWriteData].
4. If D0F0x64_x4D[ReqToggle]==0, set it to 1. If D0F0x64_x4D[ReqToggle]==1, clear it to 0.

In each sequence above, the writes the writes to D0F0x64_x4D may be combined into a single write.

#### SMUx01 MCU Config Register

Write-only. Reset: 0000_0002h.

| Bits | Description |
|---|---|
| 31:19 | Reserved. |
| 18 | **VectorOverride**. Specifies the address range used for hardware interrupts received by the microcontroller. 0=FFFFh:FFD4h. 1=BFFFh:BFD4h. |
| 17:2 | Reserved. |
| 1 | **Reset**. This field is used to place the SMU microcontroller into the reset state. 0=Microcontroller is placed into reset. 1=Microcontroller is removed from reset. |
| 0 | **RamSwitch**. Specifies whether access to SMU firmware RAM is enabled. 0=Disabled. 1=Enabled. See 2.12.1.1 [Updating Firmware]. |

#### SMUx03 MCU IRQ Register

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:11 | Reserved. |
| 10:3 | **ServiceIndex**. Read-write. Specifies the service index used by microcontroller firmware when interrupted by software. See Table 37. |

| | |
|---|---|
| 2 | **IntDone: interrupt done**. Read-only. Specifies whether the interrupt requested by writing SMUx03[IntReq] has completed. 0=Interrupt service in progress. 1=Interrupt service complete. This field is cleared by hardware when software requests an interrupt by writing to SMUx03[IntReq]. See 2.12.1.2 [Software Interrupts]. |
| 1 | **IntAck: interrupt acknowledge**. Read-only. Specifies whether the interrupt has been acknowledged by the microcontroller. 0=Not acknowledged. 1=Acknowledged. This field is cleared by hardware when software requests an interrupt by writing to SMUx03[IntReq]. See 2.12.1.2 [Software Interrupts]. |
| 0 | **IntReq: interrupt request**. Read-write. When software sets this field to 1, an interrupt is triggered to the microcontroller. No additional software interrupts can be triggered until SMUx03[IntDone]==1. Software must clear this field to 0 before setting it to 1. See 2.12.1.2 [Software Interrupts]. |

### SMUx05 SMU Data RAM

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:0 | **McuRam**. Read-write. This field is used to access SMU RAM. Reading or writing this field causes SMUx0B[MemAddr] to increment by 4. |

### SMUx0B SMU RAM Address

Reset: 0000_0000h. The index/data pair registers SMUx0B and SMUx05 are used to access the registers SMUx0B_x[8FFF:8000]. To read or write to one of these registers, the index is written first into the index register SMUx0B and then the data is read or written by reading or writing the data register SMUx05. These registers definitions represent SMU RAM locations defined for a specific purpose. GMMx100 and GMMx104 may also be used to access SMUx0B_x[8FFF:8000] by subtracting 8000h from the index before programming GMMx100.

| Bits | Description |
|---|---|
| 15:0 | **MemAddr**. Read-write. Specifies the SMU RAM address accessed. |

### SMUx0B_x830C SMU Firmware Version

Reset: xxxx_xxxxh. See 2.12.1.1 [Updating Firmware].

| Bits | Description |
|---|---|
| 31:16 | **MajorVersion**. Read-only. |
| 15:0 | **MinorVersion**. Read-only. |

### SMUx0B_x8408 RCU Power Gating Sequence 0

Reset: xxxx_xxxxh.

| Bits | Description |
|---|---|
| 31:28 | **PsoControlId7**. Read-write. |
| 27:24 | **PsoControlId6**. Read-write. |
| 23:20 | **PsoControlId5**. Read-write. |

| | |
|---|---|
| 19:16 | **PsoControlId4**. Read-write. |
| 15:12 | **PsoControlId3**. Read-write. |
| 11:8 | **PsoControlId2**. Read-write. |
| 7:4 | **PsoControlId1**. Read-write. |
| 3:0 | **PsoControlId0**. Read-write. |

**SMUx0B_x840C RCU Power Gating Sequence 1**

Reset: xxxx_xxxxh.

| Bits | Description |
|---|---|
| 31:28 | **PsoControlId15**. Read-write. |
| 27:24 | **PsoControlId14**. Read-write. |
| 23:20 | **PsoControlId13**. Read-write. |
| 19:16 | **PsoControlId12**. Read-write. |
| 15:12 | **PsoControlId11**. Read-write. |
| 11:8 | **PsoControlId10**. Read-write. |
| 7:4 | **PsoControlId9**. Read-write. |
| 3:0 | **PsoControlId8**. Read-write. |

**SMUx0B_x8410 RCU Power Gating Control**

Reset: xxxx_xxxxh.

| Bits | Description |
|---|---|
| 31:28 | **PwrGaterSel**. Read-write. |
| 27:24 | **IsoDelay**. Read-write. |
| 23:16 | **RstPulseWidth**. Read-write. |
| 15:8 | **PsoControlPeriod**. Read-write. |
| 7:3 | **PsoControlValidNum**. Read-write. |
| 2:1 | Reserved. Read-write. |
| 0 | **PwrGatingEn**. Read-write. |

**SMUx0B_x842C Alt Vdd Control**

Reset: xxxx_xxxxh.

| Bits | Description |
|---|---|
| 31:1 | Reserved. |
| 0 | **BiosAltVddE**. Read-write. |

### SMUx0B_x8434 LCLK Scaling Control

Reset: xxxx_xxxxh. See 2.5.5.1.3 [LCLK DPM].

| Bits | Description |
|---|---|
| 31:16 | **LclkTimerPeriod**. Read-write. BIOS: C350h. Specifies the time period used for LCLK DPM in REFCLKs when LCLK DPM is first enabled. Period = (4^LclkTimerPrescalar) * LclkTimerPeriod. |
| 15:8 | Reserved. |
| 7:4 | **LclkTimerPrescalar**. Read-write. BIOS: 1. See SMUx0B_x8434[LclkTimerPeriod]. |
| 3:2 | Reserved. |
| 1 | **LclkDpmType**. Read-write. BIOS: 2.5.5.1.3.3. Specifies the type of LCLK DPM used. 1=LCLK DPM using root complex activity. 0=LCLK DPM using PCIe bandwidth and DMA activity. |
| 0 | **LclkDpmEn**. Read-write. BIOS: 1. Specifies whether dynamic LCLK DPM is enabled. 1=Enabled. 0=Disabled. In addition to programming this bit, software must interrupt the SMU. |

### SMUx0B_x84[48:38:step4] LCLK DPM Sampling Period [4:0]

Reset: xxxx_xxxxh. See 2.5.5.1.3 [LCLK DPM].

**Table 123: LCLK DPM Sampling Period [4:0] to DPM State Mapping**

| Register | Bits | |
|---|---|---|
| | 31:16 | 15:0 |
| SMUx0B_x8438 | 0 | 1 |
| SMUx0B_x843C | 2 | 3 |
| SMUx0B_x8440 | 4 | 5 |
| SMUx0B_x8444 | 6 | 7 |
| SMUx0B_x8448 | 8 | 9 |

| Bits | Description |
|---|---|
| 31:16 | **FstatePeriod**. Read-write. BIOS: 2.5.5.1.3.3. See SMUx0B_x84[48:38:step4][15:0]. |
| 15:0 | **FstatePeriod**. Read-write. BIOS: 2.5.5.1.3.3. Specifies the period at which LCLK DPM samples the activity monitor and PCIe bandwidth in REFCLKs. Period = (4^SMUx0B_x8434[LclkTimerPrescalar]) * FstatePeriod. |

### SMUx0B_x84[54:4C] LCLK PCIe Up Hysteresis [2:0]

Reset: xxxx_xxxxh. See 2.5.5.1.3 [LCLK DPM].

**Table 124: LCLK PCIe Up Hysteresis [2:0] to DPM State Mapping**

| Register | Bits | | | |
|---|---|---|---|---|
| | 31:24 | 23:16 | 15:8 | 7:0 |
| SMUx0B_x844C | 0 | 1 | 2 | 3 |
| SMUx0B_x8450 | 4 | 5 | 6 | 7 |
| SMUx0B_x8454 | 8 | 9 | - | - |

| Bits | Description |
|---|---|
| 31:24 | **FstateUpHyst**. Read-write. BIOS: 0. See SMUx0B_x84[54:4C][7:0]. |
| 23:16 | **FstateUpHyst**. Read-write. BIOS: 0. See SMUx0B_x84[54:4C][7:0]. |
| 15:8 | IF (REG == SMUx0B_x8454) THEN Reserved. ELSE **FstateUpHyst**. Read-write. BIOS: 0. See SMUx0B_x84[C8:C4][7:0]. ENDIF. |
| 7:0 | IF (REG == SMUx0B_x8454) THEN Reserved. ELSE **FstateUpHyst**. Read-write. BIOS: 0. Specifies the number of times the LCLK DPM activity monitor must request a transition to a higher performance DPM state before the transition actually occurs. ENDIF. |

### SMUx0B_x84[7C:60] LCLK Activity Thresholds [7:0]

Reset: xxxx_xxxxh. The table below describes which activity threshold register corresponds to each LCLK DPM state. See 2.5.5.1.3 [LCLK DPM].

| Register | State | Register | State |
|---|---|---|---|
| SMUx0B_x8460 | 0 | SMUx0B_x8470 | 4 |
| SMUx0B_x8464 | 1 | SMUx0B_x8474 | 5 |
| SMUx0B_x8468 | 2 | SMUx0B_x8478 | 6 |
| SMUx0B_x846C | 3 | SMUx0B_x847C | 7 |

| Bits | Description |
|---|---|
| 31:16 | **Lowering**. Read-write. BIOS: 2.5.5.1.3.3. Specifies the activity threshold that the activity monitor must be under to cause LCLK DPM to transition to a lower performance state. |
| 15:0 | **Raising**. Read-write. BIOS: 2.5.5.1.3.3. Specifies the activity threshold that the activity monitor must be over to cause LCLK DPM to transition to a higher performance state. |

### SMUx0B_x84[8C:88:step4] LCLK Scaling Divisor [1:0]

Reset: xxxx_xxxxh. See 2.5.5.1.3 [LCLK DPM].

**Table 125: LCLK DPM Scaling Divisor [1:0] to DPM State Mapping**

| Register | Bits | | | |
|---|---|---|---|---|
| | 30:24 | 22:16 | 14:8 | 6:0 |
| SMUx0B_x8488 | 0 | 1 | 2 | 3 |
| SMUx0B_x848C | 4 | 5 | 6 | 7 |

| Bits | Description |
|---|---|
| 31 | Reserved. |
| 30:24 | **FstateDiv**. Read-write. BIOS: 2.5.5.1.3.3. See SMUx0B_x84[8C:88:step4][6:0]. |
| 23 | Reserved. |
| 22:16 | **FstateDiv**. Read-write. BIOS: 2.5.5.1.3.3. See SMUx0B_x84[8C:88:step4][6:0]. |
| 15 | Reserved. |

| | |
|---|---|
| 14:8 | **FstateDiv**. Read-write. BIOS: 2.5.5.1.3.3. See SMUx0B_x84[8C:88:step4][6:0]. |
| 7 | Reserved. |
| 6:0 | **FstateDiv**. Read-write. BIOS: 2.5.5.1.3.3. Specifies the LCLK divisor used when in this LCLK DPM state. See GMMx6[88:84][SclkDpmDiv]. |

### SMUx0B_x8490 RCU LCLK Scaling Control 2

Reset: xxxx_xxxxh. See 2.5.5.1.3 [LCLK DPM].

| Bits | Description |
|---|---|
| 31:24 | Reserved. |
| 23:16 | **MinDivAllowed**. Read-write. Specifies the minimum divisor supported by LCLK DPM using PCIe bandwidth and GFX DMA activity. See GMMx6[88:84][SclkDpmDiv]. |
| 15:8 | **LclkDivTtExit**. Read-write. |
| 7 | **LclkState7Valid**. Read-write. BIOS: 2.5.5.1.3.3. See SMUx0B_x8490[LclkState0Valid]. |
| 6 | **LclkState6Valid**. Read-write. BIOS: 2.5.5.1.3.3. See SMUx0B_x8490[LclkState0Valid]. |
| 5 | **LclkState5Valid**. Read-write. BIOS: 2.5.5.1.3.3. See SMUx0B_x8490[LclkState0Valid]. |
| 4 | **LclkState4Valid**. Read-write. BIOS: 2.5.5.1.3.3. See SMUx0B_x8490[LclkState0Valid]. |
| 3 | **LclkState3Valid**. Read-write. BIOS: 2.5.5.1.3.3. See SMUx0B_x8490[LclkState0Valid]. |
| 2 | **LclkState2Valid**. Read-write. BIOS: 2.5.5.1.3.3. See SMUx0B_x8490[LclkState0Valid]. |
| 1 | **LclkState1Valid**. Read-write. BIOS: 2.5.5.1.3.3. See SMUx0B_x8490[LclkState0Valid]. |
| 0 | **LclkState0Valid**. Read-write. BIOS: 2.5.5.1.3.3. Specifies whether the LCLK DPM state is valid. 1=Valid. 0=Invalid. |

### SMUx0B_x84[9C:94:step4] RCU LCLK Scaling VID [2:0]

Reset: xxxx_xxxxh. See 2.5.5.1.3 [LCLK DPM].

**Table 126: LCLK DPM Scaling VID [2:0] to DPM State Mapping**

| Register | Bits | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 27:26 | 25:24 | 19:18 | 17:16 | 11:10 | 9:8 | 3:2 | 1:0 |
| SMUx0B_x8494 | 0 | 0 | 1 | 1 | 2 | 2 | 3 | 3 |
| SMUx0B_x8498 | 4 | 4 | 5 | 5 | 6 | 6 | 7 | 7 |
| SMUx0B_x849C | 8 | 8 | 9 | 9 | - | - | - | - |

| Bits | Description |
|---|---|
| 31:28 | Reserved. |
| 27:26 | **TolExcdVid: tolerance exceeded VID**. Read-write. BIOS: 2.5.5.1.3.3. See SMUx0B_x84[9C:94:step4][3:2]. |
| 25:24 | **BaseVid**. Read-write. BIOS: 2.5.5.1.3.3. See SMUx0B_x84[9C:94:step4][1:0]. |
| 23:20 | Reserved. |
| 19:18 | **TolExcdVid: tolerance exceeded VID**. Read-write. BIOS: 2.5.5.1.3.3. See SMUx0B_x84[9C:94:step4][3:2]. |

| | |
|---|---|
| 17:16 | **BaseVid**. Read-write. BIOS: 2.5.5.1.3.3. See SMUx0B_x84[9C:94:step4][1:0]. |
| 15:12 | Reserved. |
| 11:10 | IF (REG == SMUx0B_x849C) THEN Reserved. ELSE **TolExcdVid: tolerance exceeded VID**. Read-write. BIOS: 2.5.5.1.3.3. See SMUx0B_x84[9C:94:step4][3:2]. ENDIF. |
| 9:8 | IF (REG == SMUx0B_x849C) THEN Reserved. ELSE **BaseVid**. Read-write. BIOS: 2.5.5.1.3.3. See SMUx0B_x84[9C:94:step4][1:0]. ENDIF. |
| 7:4 | Reserved. |
| 3:2 | IF (REG == SMUx0B_x849C) THEN Reserved.<br>ELSE<br>**TolExcdVid: tolerance exceeded VID**. Read-write. BIOS: 2.5.5.1.3.3. When using LCLK DPM based on root complex activity, this field is unused. When using LCLK DPM based on PCIe bandwidth and GFX activity, this field specifies the tolerance VID for the DPM state. This field indexes into D18F3x15C as follows:<br>Bits VID code — Bits VID code<br>00b D18F3x15C[SclkVidLevel0] — 10b D18F3x15C[SclkVidLevel2]<br>01b D18F3x15C[SclkVidLevel1] — 11b D18F3x15C[SclkVidLevel3]<br>ENDIF. |
| 1:0 | IF (REG == SMUx0B_x849C) THEN Reserved.<br>ELSE<br>**BaseVid**. Read-write. BIOS: 2.5.5.1.3.3. When using LCLK DPM based on root complex activity, this field specifies the VID code requested when in the DPM state. When using LCLK scaling based on PCIe bandwidth and GFX DMA activity, this field specifies the base VID for the DPM state. This field indexes into D18F3x15C as follows:<br>Bits VID code — Bits VID code<br>00b D18F3x15C[SclkVidLevel0] — 10b D18F3x15C[SclkVidLevel2]<br>01b D18F3x15C[SclkVidLevel1] — 11b D18F3x15C[SclkVidLevel3]<br>ENDIF. |

### SMUx0B_x84A0 RCU Power Gating Control 2

Reset: xxxx_xxxxh.

| Bits | Description |
|---|---|
| 31:16 | **MothPsoPwrdn**. Read-write. |
| 15:0 | **MothPsoPwrup**. Read-write. |

### SMUx0B_x84A4 RCU Power Gating Control 3

Reset: xxxx_xxxxh.

| Bits | Description |
|---|---|
| 31:16 | **DaugPsoPwrdn**. Read-write. |
| 15:0 | **DaugPsoPwrup**. Read-write. |

### SMUx0B_x84A8 RCU Power Gating Control 4

Reset: xxxx_xxxxh.

| Bits | Description |
|---|---|
| 31:16 | **IsoTimer**. Read-write. |
| 15:0 | **ResetTimer**. Read-write. |

### SMUx0B_x84[C8:C4] LCLK PCIe Down Hysteresis [1:0]

Reset: xxxx_xxxxh. See 2.5.5.1.3 [LCLK DPM].

**Table 127: LCLK PCIe Down Hysteresis [2:0] to DPM State Mapping**

| Register | Bits | | | |
|---|---|---|---|---|
| | 31:24 | 23:16 | 15:8 | 7:0 |
| SMUx0B_x84C4 | 0 | 1 | 2 | 3 |
| SMUx0B_x84C8 | 4 | 5 | 6 | 7 |
| SMUx0B_x84CC | 8 | 9 | - | - |

| Bits | Description |
|---|---|
| 31:24 | **FstateDnHyst**. Read-write. BIOS: 0. See SMUx0B_x84[C8:C4][7:0]. |
| 23:16 | **FstateDnHyst**. Read-write. BIOS: 0. See SMUx0B_x84[C8:C4][7:0]. |
| 15:8 | **FstateDnHyst**. Read-write. BIOS: 0. See SMUx0B_x84[C8:C4][7:0]. |
| 7:0 | **FstateDnHyst**. Read-write. BIOS: 0. Specifies the number of times the LCLK DPM activity monitor must request a transition to a lower performance DPM state before the transition actually occurs. |

### SMUx0B_x84[D4:D0] LCLK PCIe Scaling Tolerance [2:1]

Reset: xxxx_xxxxh. See 2.5.5.1.3 [LCLK DPM].

**Table 128: LCLK PCIe Scaling Divisor [2:0] to DPM State Mapping**

| Register | Bits | | | |
|---|---|---|---|---|
| | 30:24 | 22:16 | 14:8 | 6:0 |
| SMUx0B_x84CC | - | - | 0 | 1 |
| SMUx0B_x84D0 | 2 | 3 | 4 | 5 |
| SMUx0B_x84D4 | 6 | 7 | 8 | 9 |

| Bits | Description |
|---|---|
| 31 | Reserved. |
| 30:24 | **FstateDivTol**. Read-write. BIOS: 2.5.5.1.3.3. See SMUx0B_x84[D4:D0][6:0]. |
| 23 | Reserved. |
| 22:16 | **FstateDivTol**. Read-write. BIOS: 2.5.5.1.3.3. See SMUx0B_x84[D4:D0][6:0]. |
| 15 | Reserved. |

| | |
|---|---|
| 14:8 | **FstateDivTol**. Read-write. BIOS: 2.5.5.1.3.3. See SMUx0B_x84[D4:D0][6:0]. |
| 7 | Reserved. |
| 6:0 | **FstateDivTol**. Read-write. BIOS: 2.5.5.1.3.3. Specifies the minimum LCLK divisor (maximum frequency) supported by SMUx0B_x84[9C:94:step4][BaseVid] of the DPM state. See GMMx6[88:84][SclkDpmDiv]. |

### SMUx0B_x84[E0:D8] LCLK PCIe Scaling Divisor [2:0]

Reset: xxxx_xxxxh. See 2.5.5.1.3 [LCLK DPM].

**Table 129: LCLK PCIe Scaling Divisor [2:0] to DPM State Mapping**

| Register | Bits | | | |
|---|---|---|---|---|
| | 30:24 | 22:16 | 14:8 | 6:0 |
| SMUx0B_x84D8 | 0 | 1 | 2 | 3 |
| SMUx0B_x84DC | 4 | 5 | 6 | 7 |
| SMUx0B_x84E0 | 8 | 9 | - | - |

| Bits | Description |
|---|---|
| 31 | Reserved. |
| 30:24 | **BaseDiv**. Read-write. BIOS: 2.5.5.1.3.3. See SMUx0B_x84[E0:D8][6:0]. |
| 23 | Reserved. |
| 22:16 | **BaseDiv**. Read-write. BIOS: 2.5.5.1.3.3. See SMUx0B_x84[E0:D8][6:0]. |
| 15 | Reserved. |
| 14:8 | IF (REG == SMUx0B_x84E0) THEN Reserved. ELSE **BaseDiv**. Read-write. BIOS: 2.5.5.1.3.3. See SMUx0B_x84[E0:D8][6:0]. ENDIF. |
| 7 | Reserved. |
| 6:0 | IF (REG == SMUx0B_x84E0) THEN Reserved. ELSE **BaseDiv**. Read-write. BIOS: 2.5.5.1.3.3. Specifies the LCLK divisor used when in this LCLK DPM state. See GMMx6[88:84][SclkDpmDiv]. ENDIF. |

### SMUx0B_x84EC SMU Reset Status

Reset: xxxx_xxxxh. See 2.12.1.1 [Updating Firmware].

| Bits | Description |
|---|---|
| 31:1 | Reserved. |
| 0 | **SaveStateDone**. Read-write, updated-by-hardware. When updating SMU firmware, this field specifies when the SMU has completed saving its configuration to internal memory. 1=State save done. 0=State save incomplete. |

### SMUx0B_x8580 Power Density Multiplier Control 0

Reset: xxxx_xxxxh. See 2.5.3.1.1 [Core Performance Boost (CPB)].

| Bits | Description |
|---|---|
| 31:16 | **PdmPeriod**. Read-write. IF (BoostEn) THEN BIOS: 1388h. ENDIF. See SMUx0B_x8580[PdmUnit]. |
| 15:12 | **PdmUnit**. Read-write. IF (BoostEn) THEN BIOS: 1. ENDIF. Specifies the sampling period for power density multipliers in REFCLKs. Sampling period = (4^PdmUnit) * PdmPeriod. |
| 11 | **Reserved.** |
| 10 | **PdmCacEn: PDM cac monitoring enable**. Read-write. IF (BoostEn) THEN BIOS: 1. ENDIF. Specifies whether GPU power headroom is calculated whenever power density multipliers are enabled. 1=Calculate headroom. 0=Do not calculate headroom. |
| 9:1 | Reserved. |
| 0 | **PdmEn: PDM enable**. Read-write. IF (BoostEn && !BatteryPower) THEN BIOS: 1. ENDIF. Specifies whether power density multipliers are enabled or disabled when the SMU is interrupted with Service Index 12h is called. 1=Enabled. 0=Disabled. See 2.12.1.2 [Software Interrupts]. |

### SMUx0B_x858C PCIe Power Gating Arguments 0

Reset: xxxx_xxxxh. See 2.5.5.4.4 [Gfx Link Core Power Gating] and 2.12.1.2 [Software Interrupts].

| Bits | Description |
|---|---|
| 31:24 | **UpperLaneId**. Read-write. Specifies the highest numbered PCIe lane to mark used/unused. See Table 34. |
| 23:16 | **LowerLaneId**. Read-write. Indicates the lowest numbered PCIe lane to mark used/unused. See SMUx0B_x858C[UpperLaneId]. |
| 15:3 | Reserved. |
| 2 | **Core**. Read-write. Specifies whether hardware considers the actions taken by the SMU interrupt when determining whether to power the Gfx-link core on/off. 0=Consider this command. 1=Do not consider this command. |
| 1 | **Tx**. Read-write. Specifies whether to mark the TX side of each lane used/unused. 0=Do not change state of TX lanes. 1=Change state of TX lanes. |
| 0 | **Rx**. Read-write. Specifies whether to mark the RX side of each lane used/unused. 0=Do not change state of RX lanes. 1=Change state of RX lanes. |

### SMUx0B_x859C PCIe Power Gating Arguments 2

Reset: xxxx_xxxxh. See 2.5.5.4.4 [Gfx Link Core Power Gating] and 2.12.1.2 [Software Interrupts].

| Bits | Description |
|---|---|
| 31:1 | Reserved. |
| 0 | **PllId**. Read-write. Specifies whether software needs access to the DDI PLL within the GFX-link core. 1=Access needed. 0=No access needed. |

### SMUx0B_x85B0 PCIe PLL Wait Time

Reset: xxxx_xxxxh.

| Bits | Description |
|---|---|
| 31:8 | Reserved. |
| 7:0 | **WaitTime**. Read-write. Specifies the number of microseconds to wait for PLL relock after the PLL switches frequencies.<br>Bits: Definition<br>00h: 0 usec<br>FFh-01h: <WaitTime> usec |

### SMUx0B_x85D0 Cable Safe Configuration

Reset: xxxx_xxxxh.

| Bits | Description |
|---|---|
| 31:16 | **Period**. Read-write. Specifies the polling period for Cable Safe. The polling period = (4^Unit) * 10ns * Period * SoftPeriod. |
| 15:12 | Reserved. |
| 11:8 | **Unit**. Read-write. BIOS: 1. |
| 7:4 | **SoftPeriod**. Read-write. Specifies a firmware timer value that increments every (4^Unit) * 10ns * Period. |
| 3:1 | **DebounceFilter**. Read-write. Specifies the time period that a hot plug event has to stay valid before firmware accepts it as a hot plug event. Debounce time = (4^Unit) * 10ns * Period * Soft Period * DebounceFilter. |
| 0 | **Enable**. Read-write. 1=Enable Cable Safe polling mode. 0=Disable Cable Safe polling. |

### SMUx0B_x8600 DMA Transaction Array 1

Reset: xxxx_xxxxh.

| Bits | Description |
|---|---|
| 31:24 | **TransactionCount**. Read-write. |
| 23:16 | **MemAddr[15:8]**. Read-write. |
| 15:8 | **MemAddr[7:0]**. Read-write. |
| 7:0 | **Txn1MBusAddr[7:0]**. Read-write. |

### SMUx0B_x8604 DMA Transaction Array 2

Reset: xxxx_xxxxh.

| Bits | Description |
|---|---|
| 31:24 | **Txn1MBusAddr[15:8]**. Read-write. |
| 23:16 | **Txn1MBusAddr[23:16]**. Read-write. |

| | |
|---|---|
| 15:8 | **Txn1MBusAddr[31:24]**. Read-write. |
| 7:0 | **Txn1TransferLength[7:0]**. Read-write. |

**SMUx0B_x8608 DMA Transaction Array 3**

Reset: xxxx_xxxxh.

| Bits | Description |
|---|---|
| 31:30 | **Txn1Tsize**. Read-write. |
| 29:24 | **Txn1TransferLength[13:8]**. Read-write. |
| 23:20 | **Txn1Spare**. Read-write. |
| 19 | **Txn1Overlap**. Read-write. |
| 18 | **Txn1Static**. Read-write. |
| 17:16 | **Txn1Mode**. Read-write. |
| 15:8 | **Txn2Mbusaddr70**. Read-write. |
| 7:0 | **Txn2Mbusaddr158**. Read-write. |

**SMUx0B_x860C DMA Transaction Array 4**

Reset: xxxx_xxxxh.

| Bits | Description |
|---|---|
| 31:24 | **Txn2MBusAddr2316**. Read-write. |
| 23:16 | **Txn2MBusAddr3124**. Read-write. |
| 15:8 | **Txn2TransferLength70**. Read-write. |
| 7:6 | **Txn2Tsize**. Read-write. |
| 5:0 | **Txn2TransferLength138**. Read-write. |

**SMUx0B_x8610 DMA Transaction Array 5**

Reset: xxxx_xxxxh.

| Bits | Description |
|---|---|
| 31:28 | **Txn2Spare**. Read-write. |
| 27 | **Txn2Overlap**. Read-write. |
| 26 | **Txn2Static**. Read-write. |
| 25:24 | **Txn2Mode**. Read-write. |
| 23:16 | **Txn3MBusAddr70**. Read-write. |
| 15:8 | **Txn3MBusAddr158**. Read-write. |
| 7:0 | **Txn3MBusAddr2316**. Read-write. |

**SMUx0B_x8614 DMA Transaction Array 6**

Reset: xxxx_xxxxh.

| Bits | Description |
|---|---|
| 31:24 | **Txn3MBusAddr3124**. Read-write. |
| 23:16 | **Txn3TransferLength70**. Read-write. |
| 15:14 | **Txn3Tsize**. Read-write. |
| 13:8 | **Txn3TransferLength138**. Read-write. |
| 7:4 | **Txn3Spare**. Read-write. |
| 3 | **Txn3Overlap**. Read-write. |
| 2 | **Txn3Static**. Read-write. |
| 1:0 | **Txn3Mode**. Read-write. |

**SMUx0B_x8618 DMA Transaction Array 7**

Reset: xxxx_xxxxh.

| Bits | Description |
|---|---|
| 31:24 | **Txn4MBusAddr70**. Read-write. |
| 23:16 | **Txn4MBusAddr158**. Read-write. |
| 15:8 | **Txn4MBusAddr2316**. Read-write. |
| 7:0 | **Txn4MBusAddr3124**. Read-write. |

**SMUx0B_x861C DMA Transaction Array 8**

Reset: xxxx_xxxxh.

| Bits | Description |
|---|---|
| 31:24 | **Txn4TransferLength70**. Read-write. |
| 23:22 | **Txn4Tsize**. Read-write. |
| 21:16 | **Txn4TransferLength138**. Read-write. |
| 15:12 | **Txn4Spare**. Read-write. |
| 11 | **Txn4Overlap**. Read-write. |
| 10 | **Txn4Static**. Read-write. |
| 9:8 | **Txn4Mode**. Read-write. |
| 7:0 | **Txn5Mbusaddr70**. Read-write. |

**SMUx0B_x8620 DMA Transaction Array 9**

Reset: xxxx_xxxxh.

| Bits | Description |
|---|---|
| 31:24 | **Txn5MBusAddr158**. Read-write. |

| | |
|---|---|
| 23:16 | **Txn5MBusAddr2316**. Read-write. |
| 15:8 | **Txn5MBusAddr3124**. Read-write. |
| 7:0 | **Txn5TransferLength70**. Read-write. |

**SMUx0B_x8624 DMA Transaction Array 10**

Reset: xxxx_xxxxh.

| Bits | Description |
|---|---|
| 31:30 | **Txn5Tsize**. Read-write. |
| 29:24 | **Txn5TransferLength138**. Read-write. |
| 23:20 | **Txn5Spare**. Read-write. |
| 19 | **Txn5Overlap**. Read-write. |
| 18 | **Txn5Static**. Read-write. |
| 17:16 | **Txn5Mode**. Read-write. |
| 15:8 | **Txn6MBusAddr70**. Read-write. |
| 7:0 | **Txn6MBusAddr158**. Read-write. |

**SMUx0B_x8628 DMA Transaction Array 11**

Reset: xxxx_xxxxh.

| Bits | Description |
|---|---|
| 31:24 | **Txn6MBusAddr2316**. Read-write. |
| 23:16 | **Txn6MBusAddr3124**. Read-write. |
| 15:8 | **Txn6TransferLength70**. Read-write. |
| 7:6 | **Txn6Tsize**. Read-write. |
| 5:0 | **Txn6TransferLength138**. Read-write. |

**SMUx0B_x862C DMA Transaction Array 12**

Reset: xxxx_xxxxh.

| Bits | Description |
|---|---|
| 31:28 | **Txn6Spare**. Read-write. |
| 27 | **Txn6Overlap**. Read-write. |
| 26 | **Txn6Static**. Read-write. |
| 25:24 | **Txn6Mode**. Read-write. |
| 23:16 | **Txn7MBusAddr70**. Read-write. |
| 15:8 | **Txn7MBusAddr158**. Read-write. |
| 7:0 | **Txn7MBusAddr2316**. Read-write. |

**SMUx0B_x8630 DMA Transaction Array 13**

Reset: xxxx_xxxxh.

| Bits | Description |
|---|---|
| 31:24 | **Txn7MBusAddr3124**. Read-write. |
| 23:16 | **Txn7TransferLength70**. Read-write. |
| 15:14 | **Txn7Tsize**. Read-write. |
| 13:8 | **Txn7TransferLength138**. Read-write. |
| 7:4 | **Txn7Spare**. Read-write. |
| 3 | **Txn7Overlap**. Read-write. |
| 2 | **Txn7Static**. Read-write. |
| 1:0 | **Txn7Mode**. Read-write. |

**SMUx0B_x8634 DMA Transaction Array 14**

Reset: xxxx_xxxxh.

| Bits | Description |
|---|---|
| 31:24 | **Txn8MBusAddr70**. Read-write. |
| 23:16 | **Txn8MBusAddr158**. Read-write. |
| 15:8 | **Txn8MBusAddr2316**. Read-write. |
| 7:0 | **Txn8MBusAddr3124**. Read-write. |

**SMUx0B_x8638 DMA Transaction Array 15**

Reset: xxxx_xxxxh.

| Bits | Description |
|---|---|
| 31:24 | **Txn8TransferLength70**. Read-write. |
| 23:22 | **Txn8Tsize**. Read-write. |
| 21:16 | **Txn8TransferLength138**. Read-write. |
| 15:12 | **Txn8Spare**. Read-write. |
| 11 | **Txn8Overlap**. Read-write. |
| 10 | **Txn8Static**. Read-write. |
| 9:8 | **Txn8Mode**. Read-write. |
| 7:0 | **Txn9MBusAddr70**. Read-write. |

**SMUx0B_x863C DMA Transaction Array 16**

Reset: xxxx_xxxxh.

| Bits | Description |
|---|---|
| 31:24 | **Txn9MBusAddr158**. Read-write. |

| | |
|---|---|
| 23:16 | **Txn9MBuAaddr2316**. Read-write. |
| 15:8 | **Txn9MBusAddr3124**. Read-write. |
| 7:0 | **Txn9TransferLength70**. Read-write. |

**SMUx0B_x8640 DMA Transaction Array 17**

Reset: xxxx_xxxxh.

| Bits | Description |
|---|---|
| 31:30 | **Txn9Tsize**. Read-write. |
| 29:24 | **Txn9TransferLength138**. Read-write. |
| 23:20 | **Txn9Spare**. Read-write. |
| 19 | **Txn9Overlap**. Read-write. |
| 18 | **Txn9Static**. Read-write. |
| 17:16 | **Txn9Mode**. Read-write. |
| 15:8 | **Txn10MBusAddr70**. Read-write. |
| 7:0 | **Txn10MBusAddr158**. Read-write. |

**SMUx0B_x86[A0:50:step4] DMA Scratch Data 21-1**

Reset: xxxx_xxxxh.

| Bits | Description |
|---|---|
| 31:0 | **Data**. Read-write. |

**SMUx1B LCLK Deep Sleep Control 0**

Reset: 0000_0000h. See 2.5.5.2 [LCLK Clock Ramping (Deep Sleep)].

| Bits | Description |
|---|---|
| 15:8 | **LclkDpSlpMask: LCLK deep sleep mask**. Read-write. BIOS: IF (GpuEnabled) THEN FFh. ELSE EFh. ENDIF. Specifies which subcomponents of the RC must be idle to enter LCLK deep sleep. Each bit corresponds to a different subcomponent or state. 0=Ignore subcomponent or state. 1=Subcomponent must be idle or state must be true.<br><br>LclkDpSlpMask[7]: GFX link wrapper.<br>LclkDpSlpMask[6]: GPPMI link wrapper.<br>LclkDpSlpMask[5]: SMU microcontroller.<br>LclkDpSlpMask[4]: GFX clocks are gated.<br>LclkDpSlpMask[3]: Root complex.<br>LclkDpSlpMask[2]: NB.<br>LclkDpSlpMask[1]:<br>LclkDpSlpMask[0]: |
| 7:4 | Reserved. |

| | |
|---|---|
| 3 | **RampDis: ramping disable**. Read-write. Specifies the method used to ramp down LCLK. 0=Transition through each divisor between the current divisor and the target divisor, waiting one LCLK cycle after each transition. For example, if transitioning from /1 to /8, first transition to /2, wait one LCLK cycle, transition to /4, wait on LCLK cycle, then transition to /8. 1=Transition from the current divisor to the target divisor in a single step. |
| 2:0 | **LclkDpSlpDiv: LCLK deep sleep divisor**. Read-write. BIOS: 5. Specifies the divisor applied to LCLK in deep sleep. SMUx1D[LclkDpSlpEn] must be programmed to 0 before changing the value in this field or undefined behavior results.<br><br>Bits: 000b Definition: clocks off; Bits: 001b Definition: /2; Bits: 010b Definition: /4; Bits: 011b Definition: /8; Bits: 100b Definition: /16; Bits: 101b Definition: /32; Bits: 110b Definition: Reserved; Bits: 111b Definition: Reserved |

### SMUx1D LCLK Deep Sleep Control 1

Reset: 0000_0000h. See 2.5.5.2 [LCLK Clock Ramping (Deep Sleep)].

| Bits | Description |
|---|---|
| 15:13 | Reserved. |
| 12 | **LclkDpSlpEn: LCLK deep sleep enable**. Read-write. BIOS: 2.5.5.2.1.1. Specifies whether LCLK deep sleep is enabled. 0=Disabled. 1=Enabled. |
| 11:0 | **LclkDpSlpHyst: LCLK deep sleep hysteresis**. Read-write. BIOS: 0Fh. Specifies the hysteresis time for LCLK deep sleep in LCLK cycles. |

### SMUx33 LCLK Activity Monitor Control

Reset: 0000_0000h. See 2.5.5.1.1 [Activity Monitor].

| Bits | Description |
|---|---|
| 31:26 | Reserved. |
| 25 | **AccessCntl: access control**. Read-write. Specifies which set of registers controls the LCLK activity monitor. 0=SMUx33 and SMUx[51:35:step2]. 1=Driver specific registers. |
| 24:23 | **BusyCntSel: busy counter select**. Read-write. BIOS: 2.5.5.1.3.3. Specifies subcomponents or activity monitored by the LCLK activity monitor.<br>Bits: 00b Definition: GFX DMA (BIF); Bits: 01b Definition: All upstream activity; Bits: 10b Definition: All downstream activity; Bits: 11b Definition: ORB and IOC activity |
| 22 | **ActMonRst: activity monitor reset**. Read-write. Writing 1 to this field clears each stage of the LCLK DPM activity monitor FIR filter to 0. 1=The activity monitor is held in reset. Software must clear this bit to 0 to allow the activity monitor to function. |
| 21 | **ForceTrend**. Read-write. 1=Force the LCLK DPM activity monitor trend as specified by SMUx33[TrendMode]. 0=Do not force the LCLK DPM activity monitor trend. |
| 20 | **TrendMode**. Read-write. Specifies the trend that is forced. 0=Downward trend. 1=Upward trend. See SMUx33[ForceTrend]. |

| | |
|---|---|
| 19:16 | **LclkActMonUnt: LCLK DPM activity monitor sampling unit**. Read-write. BIOS: 2.5.5.1.3.3. See SMUx33[LclkActMonPrd]. |
| 15:0 | **LclkActMonPrd: LCLK DPM activity monitor sampling period**. Read-write. BIOS: 2.5.5.1.3.3. Specifies the sampling period for the LCLK DPM activity monitor in REFCLKs. Sampling period = (4^LclkActMonUnt) * LclkActMonPrd. |

### SMUx[51:35:step2] LCLK Activity Monitor Coefficients [14:0]

Reset: 0000_0000h. See 2.5.5.1.1 [Activity Monitor]. Each register in SMUx[51:35:step2] applies to one stage of the LCLK DPM activity monitor FIR filter as follows:

| Register | Stage | Register | Stage |
|---|---|---|---|
| SMUx35 | 0 | SMUx45 | 8 |
| SMUx37 | 1 | SMUx47 | 9 |
| SMUx39 | 2 | SMUx49 | 10 |
| SMUx3B | 3 | SMUx4B | 11 |
| SMUx3D | 4 | SMUx4D | 12 |
| SMUx3F | 5 | SMUx4F | 13 |
| SMUx41 | 6 | SMUx51 | 14 |
| SMUx43 | 7 | | |

| Bits | Description |
|---|---|
| 31:20 | Reserved. |
| 19:10 | **UpTrendCoef: upward trending coefficient**. Read-write. IF (REG==SMUx35) THEN BIOS: 24h. ELSE BIOS:22h. Specifies the coefficient used by the activity monitor FIR filter stage when the trend is upwards. |
| 9:0 | **DownTrendCoef: downward trending coefficient**. Read-write. IF (REG==SMUx35) THEN BIOS: 24h. ELSE BIOS:22h. Specifies the coefficient used by the activity monitor FIR filter stage when the trend is downwards. |

### SMUx[5D:55:step2] PCIe Lane Count Thresholds [4:0]

Reset: 0000_0000h. See 2.5.5.1.3 [LCLK DPM].

**Table 130: PCIe Lane Count Thresholds [4:0] to PCIe Bandwidth State Mapping**

| Register | Bits | |
|---|---|---|
| | 31:16 | 15:0 |
| SMUx55 | 1 | 0 |
| SMUx57 | 3 | 2 |
| SMUx59 | 5 | 4 |
| SMUx5B | 7 | 6 |
| SMUx5D | 9 | 8 |

| Bits | Description |
|---|---|
| 31:16 | **Threshold**. Read-write. See SMUx[5D:55:step2][15:0]. BIOS: 0. |
| 15:0 | **Threshold**. Read-write. BIOS: 0. Specifies the PCIe bandwidth threshold for the PCIe bandwidth state when performing LCLK DPM using PCIe bandwidth and GFX DMA activity. If the measured bandwidth is less than Threshold, the PCIe bandwidth state specified by Table 130 becomes the target state. See D0F0xE4_x013[1:0]_8031[LnCntBandwidth].<br><br>Examples:<br>• If SMUx55[15:0] == 20 (200 MB/s), SMUx55[31:16] == 50 (500 MB/s), and the measured bandwidth is 400 MB/s, then the target PCIe bandwidth state is state 0.<br>• If SMUx55[31:16] == 50 (500 MB/s), SMUx57[15:0] == 100 (1000 MB/s), and the measured bandwidth is 800 MB/s, then the target PCIe bandwidth state is state 2. |

### SMUx6F LCLK Gating Control

Reset: 0000_0000h. See 2.5.5.3 [GPU and Root Complex Clock Gating].

| Bits | Description |
|---|---|
| 31:23 | Reserved. |
| 22 | **RampDisReg**. Read-write. BIOS: 1. Specifies whether ramping of the register clock is enabled. 0=Enabled. 1=Disabled. |
| 21 | **RampDis0**. Read-write. BIOS: 1. Specifies whether ramping of dyn_oclk0 is enabled. 0=Enabled. 1=Disabled. |
| 20:12 | Reserved. |
| 11:4 | **OffDelay**. Read-write. BIOS: 1Fh. Specifies the number of SCLKs after logic goes idle before SCLK is gated off. |
| 3:0 | **OnDelay**. Read-write. Specifies the number of LCLK cycles after a transaction is received before LCLK within the SMU is ungated. |

### SMUx71 SCLK Gating Control

Reset: 0000_0000h. See 2.5.5.3 [GPU and Root Complex Clock Gating].

| Bits | Description |
|---|---|
| 31:23 | Reserved. |
| 22 | **RampDisReg**. Read-write. BIOS: 1. Specifies whether ramping of the register clock is enabled. 0=Enabled. 1=Disabled. |
| 21 | **RampDis0**. Read-write. BIOS: 1. Specifies whether ramping of dyn_oclk0 is enabled. 0=Enabled. 1=Disabled. |
| 20 | **RampDis1**. Read-write. BIOS: 1. Specifies whether ramping of dyn_oclk1 is enabled. 0=Enabled. 1=Disabled. |
| 19:12 | Reserved. |

| | |
|---|---|
| 11:4 | **OffDelay**. Read-write. BIOS: 1Fh. Specifies the number of SCLKs after logic goes idle before SCLK is gated off. |
| 3:0 | **OnDelay**. Read-write. Specifies the number of LCLK cycles after a transaction is received before LCLK within the SMU is ungated. |

**SMUx73 Clock Gating Control**

Reset: 0000_0003h. See 2.5.5.3 [GPU and Root Complex Clock Gating].

| Bits | Description |
|---|---|
| 15:2 | Reserved. |
| 1 | **DisSclkGating: enable SCLK gating**. Read-write. BIOS: 1. Specifies whether SCLK gating is enabled. 1=Disabled. 0=Enabled. |
| 0 | **DisLclkGating: enable LCLK gating**. Read-write. BIOS: 0. Specifies whether LCLK gating is enabled. 1=Disabled. 0=Enabled. |

### 3.16 Fixed Configuration Space (FCR)

See section 3.1 [Register Descriptions and Mnemonics] for a description of the register naming convention.

To read FCR space, software performs the following sequence:

1. Program the following registers:
   - SMUx0B_x8600[TransactionCount] = 1.
   - SMUx0B_x8600[MemAddr[15:8]] = 86h.
   - SMUx0B_x8600[MemAddr[7:0]] = 50h.
   - SMUx0B_x8600[Txn1MBusAddr[7:0]] = bits [7:0] of the FCR register's address.
   - SMUx0B_x8604[Txn1MBusAddr[15:8]] = bits [15:8] of the FCR register's address.
   - SMUx0B_x8604[Txn1MBusAddr[23:16]] = bits [23:16] of the FCR register's address.
   - SMUx0B_x8604[Txn1MBusAddr[31:24]] = bits [31:24] of the FCR register's address.
   - SMUx0B_x8604[Txn1TransferLength[7:0]] = 4.
   - SMUx0B_x8608[Txn1Tsize] = 3.
   - SMUx0B_x8608[Txn1TransferLenth[13:8]] = 0.
   - SMUx0B_x8608[Txn1Overlap] = 0.
   - SMUx0B_x8608[Txn1Static] = 1.
   - SMUx0B_x8608[Txn1Mode] = 0.
2. If the FCR register's address begins with FE00h, interrupt the SMU with Service Index 0Dh. If the FCR register's address begins with FF30h, interrupt the SMU with Service Index 0Bh. Wait for the interrupt to complete. See 2.12.1.2 [Software Interrupts].
3. Read data from SMUx0B_x8650 (see SMUx0B_x86[A0:50:step4]).

To write FCR space, software performs the following sequence:

1. Write data to SMUx0B_x8650 (see SMUx0B_x86[A0:50:step4]).
2. Program the following registers:
   - SMUx0B_x8600[TransactionCount] = 1.
   - SMUx0B_x8600[MemAddr[15:8]] = 86h.
   - SMUx0B_x8600[MemAddr[7:0]] = 50h.
   - SMUx0B_x8600[Txn1MBusAddr[7:0]] = bits [7:0] of the FCR register's address.
   - SMUx0B_x8604[Txn1MBusAddr[15:8]] = bits [15:8] of the FCR register's address.

- SMUx0B_x8604[Txn1MBusAddr[23:16]] = bits [23:16] of the FCR register's address.
- SMUx0B_x8604[Txn1MBusAddr[31:24]] = bits [31:24] of the FCR register's address.
- SMUx0B_x8604[Txn1TransferLength[7:0]] = 4.
- SMUx0B_x8608[Txn1Tsize] = 3.
- SMUx0B_x8608[Txn1TransferLenth[13:8]] = 0.
- SMUx0B_x8608[Txn1Overlap] = 0.
- SMUx0B_x8608[Txn1Static] = 1.
- SMUx0B_x8608[Txn1Mode] = 1.

3. Interrupt the SMU with Service Index 0Bh and wait for the interrupt to complete as described in 2.12.1.2 [Software Interrupts].
4. Read data from SMUx0B_x8650 (see SMUx0B_x86[A0:50:step4]).

**FCRxFE00_4003 VCLK Configuration 0**

| Bits | Description |
|---|---|
| 31:14 | Reserved. |
| 13:7 | **VclkDid0**. Read-only. Reset: Product-specific. |
| 6:0 | Reserved. |

**FCRxFE00_4008 VCLK Configuration 1**

| Bits | Description |
|---|---|
| 31:9 | Reserved. |
| 8:2 | **VclkDid1**. Read-only. Reset: Product-specific. |
| 1:0 | Reserved. |

**FCRxFE00_4028 VCLK Configuration 2**

| Bits | Description |
|---|---|
| 31:7 | Reserved. |
| 6:0 | **VclkDid2**. Read-only. Reset: Product-specific. |

**FCRxFE00_4036 Power Configuration Miscellaneous**

| Bits | Description |
|---|---|
| 31:15 | Reserved. |

| | |
|---|---|
| 14:13 | **PcieGen2Vid**. Read-only. Reset: Product-specific. Specifies the voltage required for gen2 PCIe operation. See 2.5.1.3 [BIOS Requirements for Power Plane Initialization]. This field indexes into D18F3x15C as follows:<br>Bits — VID code<br>00b — D18F3x15C[SclkVidLevel0]<br>01b — D18F3x15C[SclkVidLevel1]<br>10b — D18F3x15C[SclkVidLevel2]<br>11b — D18F3x15C[SclkVidLevel3] |
| 12:0 | Reserved. |

**FCRxFE00_6000 NB P-state Configuration 2**

| Bits | Description |
|---|---|
| 31:21 | Reserved. |
| 20:14 | **NbPs1Vid**. Read-only. Reset: Product-specific. Specifies the initial NBP1 voltage. See 2.5.4.1.1 [BIOS Requirements for NB P-state Initialization During DRAM Training]. |
| 13:7 | **NbPs0Vid**. Read-only. Reset: Product-specific. Specifies the initial NBP0 voltage. See 2.5.4.1.1 [BIOS Requirements for NB P-state Initialization During DRAM Training]. |
| 6:0 | Reserved. |

**FCRxFE00_6002 NB P-state Configuration 0**

| Bits | Description |
|---|---|
| 31:19 | Reserved. |
| 18:12 | **NbPs1VidHigh**. Read-only. Reset: Product-specific. Specifies a VID code used when calculating NB P-state voltages. See 2.5.4.1.1 [BIOS Requirements for NB P-state Initialization During DRAM Training]. |
| 11:5 | **NbPs1VidAddl: NBP1 VID additional**. Read-only. Reset: Product-specific. Specifies a VID code used when calculating NB P-state voltages. See 2.5.4.1.1 [BIOS Requirements for NB P-state Initialization During DRAM Training]. |
| 4:0 | Reserved. |

**FCRxFE00_600E Clock Configuration**

| Bits | Description |
|---|---|
| 31:6 | Reserved. |
| 5:0 | **MainPllOpFreqIdStartup**. Value: Product-specific. Specifies the COF of the main PLL following a cold reset. See D18F3xD4[MainPllOpFreqId]. |

**FCRxFE00_6022 DCLK/VCLK Selectors**

| Bits | Description |
|---|---|
| 31:15 | Reserved. |

| | |
|---|---|
| 14:13 | **DclkVclkSel5**. Read-only. Reset: Product-specific. |
| 12:11 | **DclkVclkSel4**. Read-only. Reset: Product-specific. |
| 10:9 | **DclkVclkSel3**. Read-only. Reset: Product-specific. |
| 8:7 | **DclkVclkSel2**. Read-only. Reset: Product-specific. |
| 6:5 | **DclkVclkSel1**. Read-only. Reset: Product-specific. |
| 4:3 | **DclkVclkSel0**. Read-only. Reset: Product-specific. |
| 2:0 | Reserved. |

### FCRxFE00_7006 NB P-state Configuration 1

| Bits | Description |
|---|---|
| 31:26 | Reserved. |
| 25:21 | **MaxNbFreqAtMinVid: maximum at NB frequency at minimum NB VID**. Read-only. Reset: Product-specific. Specifies a frequency used when calculating NB P-state frequencies. See and 2.5.4.1.1 [BIOS Requirements for NB P-state Initialization During DRAM Training].<br>Bits / Description: 00001b-00000b Reserved.; 00010b 100 MHz; 00101b-00011b Reserved.; 00110b 200 MHz; 01001b-00111b Reserved.; 01010b 267 MHz; 01101b-01011b Reserved.; 01110b 333 MHz; 10001b-01111b Reserved.; 10010b 400 MHz; 10101b-10011b Reserved.; 10110b 467 MHz; 11111b-10111b Reserved. |
| 20:14 | **NbPs1NclkDiv**. Read-only. Reset: Product-specific. Specifies the initial NBP1 frequency. See 2.5.4.1.1 [BIOS Requirements for NB P-state Initialization During DRAM Training]. |
| 13:0 | Reserved. |

### FCRxFE00_7009 NB P-state Configuration 3

| Bits | Description |
|---|---|
| 31:9 | Reserved. |
| 8:2 | **NbPs0NclkDiv**. Read-only. Reset: Product-specific. Specifies the initial NBP0 frequency. See 2.5.4.1.1 [BIOS Requirements for NB P-state Initialization During DRAM Training]. |
| 1:0 | Reserved. |

### FCRxFE00_705F GPU and RC Configuration Miscellaneous 0

| Bits | Description |
|---|---|
| 31:9 | Reserved. |

| | |
|---|---|
| 8:5 | **GnbIdleAdjustVid**. Read-only. Reset: Product-specific. Specifies the voltage offset that is subtracted from VDDNB as specified by FCRxFF30_0191[GfxIdleVoltChgMode, GfxIdleVoltChgMode]. See 2.5.5.4 [GPU and Root Complex Power Gating].<br><br>Voltage offset = GnbIdleAdjustVid * 12.5mV. |
| 4:0 | Reserved. |

**FCRxFE00_7103 SCLK DPM Configuration 0**

| Bits | Description |
|---|---|
| 31:15 | Reserved. |
| 14:13 | **SclkDpmVid4**. Read-only. Reset: Product-specific. |
| 12:11 | **SclkDpmVid3**. Read-only. Reset: Product-specific. |
| 10:9 | **SclkDpmVid2**. Read-only. Reset: Product-specific. |
| 8:7 | **SclkDpmVid1**. Read-only. Reset: Product-specific. |
| 6:5 | **SclkDpmVid0**. Read-only. Reset: Product-specific. |
| 4:0 | Reserved. |

**FCRxFE00_7104 SCLK DPM Configuration 1**

| Bits | Description |
|---|---|
| 31:28 | Reserved. |
| 27:21 | **SclkDpmDid2**. Read-only. Reset: Product-specific. |
| 20:14 | **SclkDpmDid1**. Read-only. Reset: Product-specific. |
| 13:7 | **SclkDpmDid0**. Read-only. Reset: Product-specific. |
| 6:0 | Reserved. |

**FCRxFE00_7107 SCLK DPM Configuration 2**

| Bits | Description |
|---|---|
| 31:18 | Reserved. |
| 17:11 | **SclkDpmDid4**. Read-only. Reset: Product-specific. |
| 10:4 | **SclkDpmDid3**. Read-only. Reset: Product-specific. |
| 3:0 | Reserved. |

**FCRxFE00_7109 SCLK DPM Configuration 3**

| Bits | Description |
|---|---|
| 31:10 | Reserved. |

| | |
|---|---|
| 9:2 | **SclkDpmCacBase**. Read-only. Reset: Product-specific. |
| 1:0 | Reserved. |

### FCRxFE00_710A GNB Power Reporting Configuration 0

| Bits | Description |
|---|---|
| 31:18 | Reserved. |
| 17:2 | **GpuPwrGtCac: GPU power gated cac**. Read-only. Value: Product-specific. Specifies the power headroom available when the GPU is power gated. |
| 1:0 | Reserved. |

### FCRxFE00_710D Dispclk Configuration

| Bits | Description |
|---|---|
| 31:30 | Reserved. |
| 29:23 | **DispclkDid3**. Read-only. Reset: Product-specific. |
| 22:16 | **DispclkDid2**. Read-only. Reset: Product-specific. |
| 15:9 | **DispclkDid1**. Read-only. Reset: Product-specific. |
| 8:2 | **DispclkDid0**. Read-only. Reset: Product-specific. |
| 1:0 | Reserved. |

### FCRxFE00_7110 LCLK DPM Configuration 0

| Bits | Description |
|---|---|
| 31:27 | Reserved. |
| 26:20 | **LclkDpmDid2**. Read-only. Reset: Product-specific. See FCRxFE00_7113[LclkDpmDid0]. |
| 19:13 | **LclkDpmDid1**. Read-only. Reset: Product-specific. See FCRxFE00_7113[LclkDpmDid0]. |
| 12:6 | **LclkDpmDid0**. Read-only. Reset: Product-specific. Specifies the divisor used for the respective LCLK DPM state. See 2.5.5.1.3.3 [BIOS Initialization for LCLK DPM Using Root Complex Activity]. |
| 5:0 | Reserved. |

### FCRxFE00_7113 LCLK DPM Configuration 1

| Bits | Description |
|---|---|
| 31:14 | Reserved. |
| 13 | **LclkDpmValid3**. Read-only. Reset: Product-specific. See FCRxFE00_7113[LclkDpmValid0]. |
| 12 | **LclkDpmValid2**. Read-only. Reset: Product-specific. See FCRxFE00_7113[LclkDpmValid0]. |
| 11 | **LclkDpmValid1**. Read-only. Reset: Product-specific. See FCRxFE00_7113[LclkDpmValid0]. |

| | |
|---|---|
| 10 | **LclkDpmValid0**. Read-only. Reset: Product-specific. Specifies whether the respective LclkDpm-DidX is valid. See 2.5.5.1.3.3 [BIOS Initialization for LCLK DPM Using Root Complex Activity]. |
| 9:3 | **LclkDpmDid3**. Read-only. Reset: Product-specific. See FCRxFE00_7113[LclkDpmDid0]. |
| 2:0 | Reserved. |

**FCRxFE00_7114 DCLK Configuration 0**

| Bits | Description |
|---|---|
| 31:27 | Reserved. |
| 26:20 | **DclkDid2**. Read-only. Reset: Product-specific. |
| 19:13 | **DclkDid1**. Read-only. Reset: Product-specific. |
| 12:6 | **DclkDid0**. Read-only. Reset: Product-specific. |
| 5:0 | Reserved. |

**FCRxFE00_7117 VCLK/DCLK Configuration 1**

| Bits | Description |
|---|---|
| 31:17 | Reserved. |
| 16:10 | **VclkDid3**. Read-only. Reset: Product-specific. |
| 9:3 | **DclkDid3**. Read-only. Reset: Product-specific. |
| 2:0 | Reserved. |

**FCRxFE00_7119 SCLK DPM Configuration 5**

| Bits | Description |
|---|---|
| 31 | Reserved. |
| 30:26 | **SclkDpmValid5**. Read-only. Reset: Product-specific. |
| 25:21 | **SclkDpmValid4**. Read-only. Reset: Product-specific. |
| 20:16 | **SclkDpmValid3**. Read-only. Reset: Product-specific. |
| 15:11 | **SclkDpmValid2**. Read-only. Reset: Product-specific. |
| 10:6 | **SclkDpmValid1**. Read-only. Reset: Product-specific. |
| 5:1 | **SclkDpmValid0**. Read-only. Reset: Product-specific. |
| 0 | Reserved. |

**FCRxFE00_711C Power Policy Labels**

| Bits | Description |
|---|---|
| 31:19 | Reserved. |
| 18:17 | **PolicyLabel5**. Read-only. Reset: Product-specific. |
| 16:15 | **PolicyLabel4**. Read-only. Reset: Product-specific. |

| | |
|---|---|
| 14:13 | **PolicyLabel3**. Read-only. Reset: Product-specific. |
| 12:11 | **PolicyLabel2**. Read-only. Reset: Product-specific. |
| 10:9 | **PolicyLabel1**. Read-only. Reset: Product-specific. |
| 8:7 | **PolicyLabel0**. Read-only. Reset: Product-specific. |
| 6:0 | Reserved. |

### FCRxFE00_711E Power Policy Flags 0

| Bits | Description |
|---|---|
| 31 | Reserved. |
| 30:24 | **PolicyFlags3**. Read-only. Reset: Product-specific. |
| 23:17 | **PolicyFlags2**. Read-only. Reset: Product-specific. |
| 16:10 | **PolicyFlags1**. Read-only. Reset: Product-specific. |
| 9:3 | **PolicyFlags0**. Read-only. Reset: Product-specific. |
| 2:0 | Reserved. |

### FCRxFE00_7121 Power Policy Flags 1

| Bits | Description |
|---|---|
| 31:21 | Reserved. |
| 20:14 | **PolicyFlags5**. Read-only. Reset: Product-specific. |
| 13:7 | **PolicyFlags4**. Read-only. Reset: Product-specific. |
| 6:0 | Reserved. |

### FCRxFF30_0134 DCCG Clock Branch Control

| Bits | Description |
|---|---|
| 31:0 | Alias of GMMx4D0. |

### FCRxFF30_0191 SCLK Power Management Control

Reset: 0000_F812h.

| Bits | Description |
|---|---|
| 31:18 | Reserved. |
| 17 | **GfxIdleVoltChgMode: GFX idle voltage change mode**. Read-write. Specifies whether the GPU must be power gated or clock gated before hardware can reduce the VDDNB voltage. 0=Power gated. 1=Clock gated. See 2.5.1.4.2 [Alternate Low Power Voltages]. |

| | |
|---|---|
| 16 | **GfxIdleVoltChgEn: GFX idle voltage change enable**. Read-write; updated-by-hardware. Specifies whether hardware reduces the VDDNB voltage when the GPU is clock gated or power gated. 0=No voltage reduction. 1=Reduce voltage as specified by FCRxFF30_0191[GfxIdleVoltChgMode] and FCRxFE00_705F[GnbIdleAdjustVid]. See 2.5.1.4.2 [Alternate Low Power Voltages]. |
| 15:0 | Reserved. |

**FCRxFF30_01E4 CG DS Voltage Control**

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:21 | Reserved. |
| 20 | **VoltageChangeEn**. Read-write. BIOS: 1. Specifies whether LCLK DPM voltage changes are enabled. 1=Enabled. 0=Disabled. See 2.5.5.1.3 [LCLK DPM]. |
| 19:0 | Reserved. |

**FCRxFF30_01F4 Clock Gating Override 0**

Reset: FFFF_FFFFh. See 2.5.5.3 [GPU and Root Complex Clock Gating].

| Bits | Description |
|---|---|
| 31 | **ReservedCgtt31Override**. Read-write. IF (!GpuEnabled) THEN BIOS: 0. ENDIF. |
| 30 | **ReservedCgtt30Override**. Read-write. IF (!GpuEnabled) THEN BIOS: 0. ENDIF. |
| 29 | **ReservedCgtt29Override**. Read-write. IF (!GpuEnabled) THEN BIOS: 0. ENDIF. |
| 28 | **CgMcdwCgttSclkOverride**. Read-write. BIOS: 2.14.2.1. |
| 27 | **CgMcbCgttSclkOverride**. Read-write. BIOS: 2.14.2.1. |
| 26 | **ReservedCgtt26Override**. Read-write. IF (!GpuEnabled) THEN BIOS: 0. ENDIF. |
| 25 | **CgDcCgttSclkOverride**. Read-write. BIOS: 0. |
| 24 | **CgDrmCgttSclkOverride**. Read-write. IF (!GpuEnabled) THEN BIOS: 0. ENDIF. |
| 23 | **CgRomCgttSclkOverride**. Read-write. IF (!GpuEnabled) THEN BIOS: 0. ENDIF. |
| 22 | **CgBifCgttSclkOverride**. Read-write. BIOS: 0. |
| 21 | **CgUvdmCgttDclkOverride**. Read-write. IF (!GpuEnabled) THEN BIOS: 0. ENDIF. |
| 20 | **CgUvdmCgttVclkOverride**. Read-write. IF (!GpuEnabled) THEN BIOS: 0. ENDIF. |
| 19 | **CgUvdmCgttSclkOverride**. Read-write. IF (!GpuEnabled) THEN BIOS: 0. ENDIF. |
| 18 | **CgAvpCgttEclkOverride**. Read-write. IF (!GpuEnabled) THEN BIOS: 0. ENDIF. |
| 17 | **CgAvpCgttSclkOverride**. Read-write. IF (!GpuEnabled) THEN BIOS: 0. ENDIF. |
| 16 | **CgVcCgttSclkOverride**. Read-write. IF (!GpuEnabled) THEN BIOS: 0. ENDIF. |
| 15 | **ReservedCgtt15Override**. Read-write. IF (!GpuEnabled) THEN BIOS: 0. ENDIF. |
| 14 | **ReservedCgtt14Override**. Read-write. IF (!GpuEnabled) THEN BIOS: 0. ENDIF. |
| 13 | **CgDb1CgttSclkOverride**. Read-write. IF (!GpuEnabled) THEN BIOS: 0. ENDIF. |
| 12 | **CgDb0CgttSclkOverride**. Read-write. IF (!GpuEnabled) THEN BIOS: 0. ENDIF. |
| 11 | **ReservedCgtt11Override**. Read-write. IF (!GpuEnabled) THEN BIOS: 0. ENDIF. |
| 10 | **ReservedCgtt10Override**. Read-write. IF (!GpuEnabled) THEN BIOS: 0. ENDIF. |

| | |
|---|---|
| 9 | **CgCb1CgttSclkOverride**. Read-write. IF (!GpuEnabled) THEN BIOS: 0. ENDIF. |
| 8 | **CgCb0CgttSclkOverride**. Read-write. IF (!GpuEnabled) THEN BIOS: 0. ENDIF. |
| 7 | **CgSxsCgttSclkOverride**. Read-write. IF (!GpuEnabled) THEN BIOS: 0. ENDIF. |
| 6 | **CgSxmCgttSclkOverride**. Read-write. IF (!GpuEnabled) THEN BIOS: 0. ENDIF. |
| 5 | **CgSpimCgttSclkOverride**. Read-write. IF (!GpuEnabled) THEN BIOS: 0. ENDIF. |
| 4 | **CgScCgttSclkOverride**. Read-write. IF (!GpuEnabled) THEN BIOS: 0. ENDIF. |
| 3 | **CgPaCgttSclkOverride**. Read-write. IF (!GpuEnabled) THEN BIOS: 0. ENDIF. |
| 2 | **CgVgtCgttSclkOverride**. Read-write. IF (!GpuEnabled) THEN BIOS: 0. ENDIF. |
| 1 | **CgCpCgttSclkOverride**. Read-write. IF (!GpuEnabled) THEN BIOS: 0. ENDIF. |
| 0 | **CgRlcCgttSclkOverride**. Read-write. IF (!GpuEnabled) THEN BIOS: 0. ENDIF. |

### FCRxFF30_01F5 Clock Gating Override 1

Reset: FFFF_FFFFh. See 2.5.5.3 [GPU and Root Complex Clock Gating].

| Bits | Description |
|---|---|
| 31 | **CgSpimCgtsSclkLsOverride**. Read-write. IF (!GpuEnabled) THEN BIOS: 0. ENDIF. |
| 30 | **CgSpimCgtsSclkOverride**. Read-write. IF (!GpuEnabled) THEN BIOS: 0. ENDIF. |
| 29 | **CgXbrCgttSclkOverride**. Read-write. IF (!GpuEnabled) THEN BIOS: 0. ENDIF. |
| 28 | **CgDcCgttDispclkOverride**. Read-write. BIOS: 0. |
| 27 | **CgUvduCgttDclkOverride**. Read-write. IF (!GpuEnabled) THEN BIOS: 0. ENDIF. |
| 26 | **CgUvduCgttVclkOverride**. Read-write. IF (!GpuEnabled) THEN BIOS: 0. ENDIF. |
| 25 | **CgUvduCgttSclkOverride**. Read-write. IF (!GpuEnabled) THEN BIOS: 0. ENDIF. |
| 24 | **CgDrmdmaCgttSclkOverride**. Read-write. IF (!GpuEnabled) THEN BIOS: 0. ENDIF. |
| 23 | **CgSrbmCgttSclkOverride**. Read-write. IF (!GpuEnabled) THEN BIOS: 0. ENDIF. |
| 22 | **CgSemCgttSclkOverride**. Read-write. IF (!GpuEnabled) THEN BIOS: 0. ENDIF. |
| 21 | **CgDbgCgttSclkOverride**. Read-write. IF (!GpuEnabled) THEN BIOS: 0. ENDIF. |
| 20 | **CgIhCgttSclkOverride**. Read-write. IF (!GpuEnabled) THEN BIOS: 0. ENDIF. |
| 19 | **ReservedCgtt51Override**. Read-write. IF (!GpuEnabled) THEN BIOS: 0. ENDIF. |
| 18 | **CgSmuCgttSclkOverride**. Read-write. IF (!GpuEnabled) THEN BIOS: 0. ENDIF. |
| 17 | **ReservedCgtt49Override**. Read-write. IF (!GpuEnabled) THEN BIOS: 0. ENDIF. |
| 16 | **CgGrbmCgttSclkOverride**. Read-write. IF (!GpuEnabled) THEN BIOS: 0. ENDIF. |
| 15 | **CgIocCgttLclkOverride**. Read-write. BIOS: 0. |
| 14 | **CgIocCgttSclkOverride**. Read-write. BIOS: 0. |
| 13 | **CgOrbCgttLclkOverride**. Read-write. BIOS: 0. |
| 12 | **CgOrbCgttSclkOverride**. Read-write. BIOS: 0. |
| 11 | **CgVmcCgttSclkOverride**. Read-write. BIOS: 2.14.2.1. |
| 10 | **CgHdpCgttSclkOverride**. Read-write. IF (!GpuEnabled) THEN BIOS: 0. ENDIF. |
| 9 | **CgSqCgttSclkOverride**. Read-write. IF (!GpuEnabled) THEN BIOS: 0. ENDIF. |
| 8 | **CgTccCgttSclkOverride**. Read-write. IF (!GpuEnabled) THEN BIOS: 0. ENDIF. |

| | |
|---|---|
| 7 | **CgTcpCgttSclkOverride**. Read-write. IF (!GpuEnabled) THEN BIOS: 0. ENDIF. |
| 6 | **CgTcaCgttSclkOverride**. Read-write. IF (!GpuEnabled) THEN BIOS: 0. ENDIF. |
| 5 | **CgTdCgttSclkOverride**. Read-write. IF (!GpuEnabled) THEN BIOS: 0. ENDIF. |
| 4 | **CgTaCgttSclkOverride**. Read-write. IF (!GpuEnabled) THEN BIOS: 0. ENDIF. |
| 3 | **ReservedCgtt35Override**. Read-write. IF (!GpuEnabled) THEN BIOS: 0. ENDIF. |
| 2 | **ReservedCgtt34Override**. Read-write. IF (!GpuEnabled) THEN BIOS: 0. ENDIF. |
| 1 | **ReservedCgtt33Override**. Read-write. IF (!GpuEnabled) THEN BIOS: 0. ENDIF. |
| 0 | **ReservedCgtt32Override**. Read-write. IF (!GpuEnabled) THEN BIOS: 0. ENDIF. |

**FCRxFF30_0398 SRBM Soft Reset Generation**

| Bits | Description |
|---|---|
| 31:24 | Reserved. |
| 23 | **SoftResetOrb**. Read-write. Reset: 0. See: SoftResetBif. |
| 22 | **SoftResetRegbb**. Read-write. Reset: 0. See: SoftResetBif. |
| 21 | **SoftResetTst**. Read-write. Reset: 0. See: SoftResetBif. |
| 20 | **SoftResetDrmdma**. Read-write. Reset: 0.See: SoftResetBif. |
| 19 | Reserved. |
| 18 | **SoftResetUvd**. Read-write. Reset: 0. BIOS: 2.12.2. See: SoftResetBif. |
| 17 | **SoftResetVmc**. Read-write. Reset: 0. See: SoftResetBif. |
| 16 | Reserved. |
| 15 | **SoftResetSem**. Read-write. Reset: 0. See: SoftResetBif. |
| 14 | **SoftResetRom**. Read-write. Reset: 0. See: SoftResetBif. |
| 13 | **SoftResetRlc**. Read-write. Reset: 0. BIOS: 2.12.2. See: SoftResetBif. |
| 12 | Reserved. |
| 11 | **SoftResetMc**. Read-write. Reset: 0. BIOS: 2.12.2. See: SoftResetBif. |
| 10 | **SoftResetIh**. Read-write. Reset: 0. See: SoftResetBif. |
| 9 | **SoftResetHdp**. Read-write. Reset: 0. See: SoftResetBif. |
| 8 | **SoftResetGrbm**. Read-write. Reset: 0. BIOS: 2.12.2. See: SoftResetBif. |
| 7 | **SoftResetDrm**. Read-write. Reset: 0. See: SoftResetBif. |
| 6 | Reserved. |
| 5 | **SoftResetDc**. Read-write. Reset: 0. BIOS: 2.12.2. See: SoftResetBif. |
| 4:3 | Reserved. |
| 2 | **SoftResetCg**. Read-write. Reset: 0. See: SoftResetBif. |
| 1 | **SoftResetBif**. Read-write. Reset: 0. 1=Soft Reset to indicated block. |
| 0 | Reserved. |

#### FCRxFF30_0AE6 GMC Miscellaneous Controls

| Bits | Description |
|---|---|
| 31:0 | Alias of GMMx2B98. |

#### FCRxFF30_1512 GPU PCI Interface Clock Gating Control

See 2.5.5.3 [GPU and Root Complex Clock Gating].

| Bits | Description |
|---|---|
| 31 | **SoftOverride0**. Read-write. Reset: 1. BIOS: 0. Soft OVERRIDE for reg_oclk. |
| 30:0 | Reserved. |

#### FCRxFF30_1529 GPU PCI Interface Busy Delay Counter

See 2.5.5.3 [GPU and Root Complex Clock Gating].

| Bits | Description |
|---|---|
| 31:6 | Reserved. |
| 5:0 | **DelayCnt**. Read-write. Reset: 3Fh. Number of clock cycles to delay the de-assertion of the GPU PCI interface busy signal. |

#### FCRxFF30_1[E,B]7C Display Controller Front-end (DCFE) Clock Control

| Bits | Description |
|---|---|
| 31:0 | Alias of GMMx[79,6D]F0. |

### 3.17 GPU Memory Mapped Registers

See 3.1 [Register Descriptions and Mnemonics] for a description of the register naming convention.

#### GMMx00 Memory Mapped Index

Reset: 0000_0000h. General Memory Access. The GMMx00 and GMMx04 pair of registers are used to indirectly accessed all other memory mapped registers in the lower 64KB space and the Frame buffer.

| Bits | Description |
|---|---|
| 31 | **Aper**. Read-write. Selects the aperture the offset is for. 0=Register aperture. 1=Linear aperture. |
| 30:0 | **Offset**. Read-write. Specifies the offset to be accessed. All accesses must be dword aligned, therefore, bits 1:0 are tied to zero. |

#### GMMx04 Memory Mapped Data

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:0 | **Data**. Read-write. Data read from or written to the offset specified in GMMx00. |

### 3.17.1 Memory Mapped SMU Registers

#### GMMx100 RCU Indirect Index Register

Reset: 0000_0000h. The index/data pair registers GMMx100 and GMMx104 are used to access the registers GMMx104_x[FFF:000]. To read or write to one of these registers, the address is written first into the address register GMMx100 and then the data is read or written by reading or writing the data register GMMx104. The index/data pair registers SMUx0B and SMUx05 can also be used to access GMMx104_x[FFF:000] by adding 8000h to the index before programming SMUx0B. See SMUx0B_x[8FFF:8000] for the register definitions.

| Bits | Description |
|---|---|
| 31:0 | **RcuIndIndex: RCU indirect index**. Read-write. |

#### GMMx104 RCU Indirect Data Register

See GMMx100.

#### GMMx4D0 DCCG Clock Branch Control

Reset: 0010_0375h. See 2.5.5.3 [GPU and Root Complex Clock Gating].

| Bits | Description |
|---|---|
| 31:10 | Reserved. |
| 9 | **SymclkbGateDisable**. Read-write. BIOS: 0. 1=Disable symbol clock B gating. |
| 8 | **SymclkaGateDisable**. Read-write. BIOS: 0.1=Disable symbol clock A gating. |
| 7:3 | Reserved. |
| 2 | **SclkGateDisable**. Read-write. BIOS: 0. 1=Disable SCLK gating for all branches in DCCG/DCI/DCO/DCFE. |
| 1 | **DispclkRDccgGateDisable**. Read-write. BIOS: 0. 1=Disable DISPCLK_R_DCCG gating. |
| 0 | **DispclkDccgGateDisable**. Read-write. BIOS: 0. 1=Disable DISPCLK gating for all branches in DCCG/DCI/DCO/DCFE. |

#### GMMx670 GPU Control

| Bits | Description |
|---|---|
| 31:2 | Reserved. |
| 1 | **GpuDis: GPU disable**. Read-only. Reset: Product-specific. Specifies whether the GPU is disabled. 0=GPU is enabled. 1=GPU is disabled. |
| 0 | Reserved. |

#### GMMx6[88:84] SCLK DPM Control [1:0]

Reset: 8F8F_8F8Fh.

**Table 131: SCLK DPM Control [1:0] to DPM State Mapping**

| Register | Bits | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 31 | 30:24 | 23 | 22:16 | 15 | 14:8 | 7 | 6:0 |
| GMMx684 | 3 | 3 | 2 | 2 | 1 | 1 | 0 | 0 |
| GMMx688 | 7 | 7 | 6 | 6 | 5 | 5 | 4 | 4 |

| Bits | Description |
|---|---|
| 31 | **SclkDpmVld**. Read-write. See: GMMx6[88:84][7]. |
| 30:24 | **SclkDpmDiv**. Read-write. See: GMMx6[88:84][6:0]. |
| 23 | **SclkDpmVld**. Read-write. See: GMMx6[88:84][7]. |
| 22:16 | **SclkDpmDiv**. Read-write. See: GMMx6[88:84][6:0]. |
| 15 | **SclkDpmVld**. Read-write. See: GMMx6[88:84][7]. |
| 14:8 | **SclkDpmDiv**. Read-write. See: GMMx6[88:84][6:0]. |
| 7 | **SclkDpmVld: DPM state valid**. Read-write. Specifies whether the SCLK DPM state is valid. 0=Not valid. 1=Valid. See 2.5.5.1.2 [SCLK DPM]. |
| 6:0 | **SclkDpmDiv: DPM state divisor**. Read-write. Specifies the divisor used when the processor is in the SCLK DPM state. See 2.5.5.1.2 [SCLK DPM].<br><br>• The clock divisor can be calculated using the following table:<br>Bits — Resulting Clock Divisor<br>00h — clocks off<br>07h-01h — reserved<br>3Fh-08h — <SclkDpmDiv* 0.25><br>5Fh-40h — <((SclkDpmDiv - 64) * 0.5) + 16><br>7Eh-60h — <((SclkDpmDiv- 96) * 1.0) + 32><br>7Fh — /128<br>• Divisor examples:<br>  • Example #1: If GMMx6[88:84][SclkDpmDiv] = 0Ch = 12d, then the divisor = 12 * 0.25 = /3.0<br>  • Example #2: If GMMx6[88:84][SclkDpmDiv] = 41h = 65d, then the divisor = (65 - 64) * 0.5 + 16 = /16.5<br>• The table above effectively allows for the following divisors to be created:<br>  • Clocks off<br>  • /2.0 to /16.0 in 0.25 steps<br>  • /16.0 to /32 in 0.50 steps<br>  • /32 to /62 in 1.00 steps<br>  • /128<br>• The COF can be calculated using the following equation:<br>  • COF = (main PLL frequency specified by D18F3xD4[MainPllOpFreqId]) / clock divisor.<br>  • Example: If D18F3xD4[MainPllOpFreqId] = 10h = 3.2GHz and GMMx6[88:84][SclkDpmDiv] = 41h = /16.5, then the SCLK COF = 193.93MHz. |

### GMMx690 Activity Monitor Voting

Reset: 0FFF_FFFFh. See 2.5.5.1.1 [Activity Monitor].

| Bits | Description |
|---|---|
| 31:20 | Reserved. |
| 19 | Reserved. |
| 18 | **UvdFreqThrottlingVoteEn**. Read-write. See GMMx690[GfxFreqThrottlingVoteEn]. |
| 17 | Reserved. |
| 16 | **DrmDmaFreqThrottlingVoteEn**. Read-write. See GMMx690[GfxFreqThrottlingVoteEn]. |
| 15 | Reserved. |
| 14 | **AvpFreqThrottlingVoteEn**. Read-write. Unused. |
| 13 | Reserved. |
| 12 | **DrmFreqThrottlingVoteEn**. Read-write. See GMMx690[GfxFreqThrottlingVoteEn]. |
| 11 | Reserved. |
| 10 | **PdmaFreqThrottlingVoteEn**. Read-write. Unused. |
| 9 | Reserved. |
| 8 | **IhSemFreqThrottlingVoteEn**. Read-write. See GMMx690[GfxFreqThrottlingVoteEn]. |
| 7 | Reserved. |
| 6 | **RomFreqThrottlingVoteEn**. Read-write. Unused. |
| 5 | Reserved. |
| 4 | **HdpFreqThrottlingVoteEn**. Read-write. See GMMx690[GfxFreqThrottlingVoteEn]. |
| 3 | Reserved. |
| 2 | **BifFreqThrottlingVoteEn**. Read-write. See GMMx690[GfxFreqThrottlingVoteEn]. |
| 1 | Reserved. |
| 0 | **GfxFreqThrottlingVoteEn**. Read-write. Specifies whether the component is included when calculating activity levels for SCLK DPM. 0=Included. 1=Not included. See 2.5.5.1.1 [Activity Monitor]. |

### GMMx6[CC:94] SCLK FIR Filter Tap Coefficients

Reset: 0000_0000h. See 2.5.5.1.1 [Activity Monitor]. Each register in GMMx6[CC:94] applies to one stage of the SCLK DPM activity monitor FIR filter as follows:

| Register | Stage | Register | Stage |
|---|---|---|---|
| GMMx694 | 0 | GMMx6B4 | 8 |
| GMMx698 | 1 | GMMx6B8 | 9 |
| GMMx69C | 2 | GMMx6BC | 10 |
| GMMx6A0 | 3 | GMMx6C0 | 11 |
| GMMx6A4 | 4 | GMMx6C4 | 12 |
| GMMx6A8 | 5 | GMMx6C8 | 13 |
| GMMx6AC | 6 | GMMx6CC | 14 |
| GMMx6B0 | 7 | | |

| Bits | Description |
|---|---|
| 31:20 | Reserved. |
| 19:10 | **UpTrendCoef: upward trending coefficient**. Read-write. Specifies the coefficient used by the appropriate activity monitor FIR filter stage when the trend is upwards. |
| 9:0 | **DownTrendCoef: downward trending coefficient**. Read-write. Specifies the coefficient used by the appropriate activity monitor FIR filter stage when the trend is downwards. |

### GMMx6E0 DPM Control 2

**Table 132: SclkDpmGnbSlow to SCLK DPM State Mapping**

| Register | Bits | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 |
| GMMx6E0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

| Bits | Description |
|---|---|
| 31 | Reserved. |
| 30 | **SclkDpmGnbSlow**. Read-write. Reset: 0. See GMMx6E0[23]. |
| 29 | **SclkDpmGnbSlow**. Read-write. Reset: 0. See GMMx6E0[23]. |
| 28 | **SclkDpmGnbSlow**. Read-write. Reset: 0. See GMMx6E0[23]. |
| 27 | **SclkDpmGnbSlow**. Read-write. Reset: 0. See GMMx6E0[23]. |
| 26 | **SclkDpmGnbSlow**. Read-write. Reset: 0. See GMMx6E0[23]. |
| 25 | **SclkDpmGnbSlow**. Read-write. Reset: 0. See GMMx6E0[23]. |
| 24 | **SclkDpmGnbSlow**. Read-write. Reset: 0. See GMMx6E0[23]. |
| 23 | **SclkDpmGnbSlow**. Read-write. Reset: 0. Specifies whether CNB transitions occur in the respective SCLK DPM state. See 2.5.5.1.2 [SCLK DPM] and 2.5.4.1 [NB P-states]. |
| 22:19 | Reserved. |
| 18 | **SclkDpmEn: DPM enable**. Read-write. Reset: 0. Specifies whether DPM is enabled. 1=Enabled. 0=Disabled. See 2.5.5.1.2 [SCLK DPM]. |
| 17:0 | Reserved. |

### GMMx71C DPM Control 3

Reset: 261A_8000h.

| Bits | Description |
|---|---|
| 31:8 | Reserved. |
| 7 | **Dpm7Wm: DPM state 7 watermark**. Read-write. See: GMMx71C[Dpm0Wm]. |
| 6 | **Dpm6Wm: DPM state 6 watermark**. Read-write. See: GMMx71C[Dpm0Wm]. |
| 5 | **Dpm5Wm: DPM state 5 watermark**. Read-write. See: GMMx71C[Dpm0Wm]. |
| 4 | **Dpm4Wm: DPM state 4 watermark**. Read-write. See: GMMx71C[Dpm0Wm]. |
| 3 | **Dpm3Wm: DPM state 3 watermark**. Read-write. See: GMMx71C[Dpm0Wm]. |

| | |
|---|---|
| 2 | **Dpm2Wm: DPM state 2 watermark**. Read-write. See: GMMx71C[Dpm0Wm]. |
| 1 | **Dpm1Wm: DPM state 1 watermark**. Read-write. See: GMMx71C[Dpm0Wm]. |
| 0 | **Dpm0Wm: DPM state 0 watermark**. Read-write. Specifies which watermark the respective DPM state should use. 0=Use the normal watermark settings. 1=Use the low watermark settings. See 2.5.5.1.2 [SCLK DPM]. |

### GMMx720 DPM Control 4

Reset: 0008_C350h.

| Bits | Description |
|---|---|
| 31:3 | Reserved. |
| 2:0 | **DpmStartState**. Read-write. Specifies the SCLK DPM state hardware transitions to when SCLK DPM is enabled. See 2.5.5.1.2 [SCLK DPM]. |

### GMMx724 DPM Control 6

Reset: 00B6_DB6Dh. See 2.5.5.2 [LCLK Clock Ramping (Deep Sleep)].

**Table 133: SCLK deep sleep to DPM state mapping**

| Register | Bits | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 23:21 | 20:18 | 17:15 | 14:12 | 11:9 | 8:6 | 5:3 | 2:0 |
| GMMx724 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

| Bits | Description |
|---|---|
| 31:24 | Reserved. |
| 23:21 | **FstateDsDivId: DPM state shallow sleep divisor**. Read-write. See GMMx724[2:0]. |
| 20:18 | **FstateDsDivId: DPM state shallow sleep divisor**. Read-write. See GMMx724[2:0]. |
| 17:15 | **FstateDsDivId: DPM state shallow sleep divisor**. Read-write. See GMMx724[2:0]. |
| 14:12 | **FstateDsDivId: DPM state shallow sleep divisor**. Read-write. See GMMx724[2:0]. |
| 11:9 | **FstateDsDivId: DPM state shallow sleep divisor**. Read-write. See GMMx724[2:0]. |
| 8:6 | **FstateDsDivId: DPM state shallow sleep divisor**. Read-write. See GMMx724[2:0]. |
| 5:3 | **FstateDsDivId: DPM state shallow sleep divisor**. Read-write. See GMMx724[2:0]. |
| 2:0 | **FstateDsDivId: DPM state shallow sleep divisor**. Read-write. Specifies the deep sleep divisor used when in the respective DPM state.<br>Bits Divisor<br>0h Clock off.<br>1h Divide by 2.<br>2h Divide by 4.<br>3h Divide by 8.<br>Bits Divisor<br>4h Divide by 16.<br>5h Divide by 32.<br>7h-6h Reserved. |

### GMMx7[48:38,30:28] DPM Activity Thresholds 7-0

Reset: 0000_0000h. See 2.5.5.1.1 [Activity Monitor] and 2.5.5.1.2 [SCLK DPM].

| Bits | Description |
|---|---|
| 31:16 | **Lowering**. Read-write. When the aggregate activity from the activity monitor falls below this threshold, DPM transitions to the next lower performance state. |
| 15:0 | **Raising**. Read-write. When the aggregate activity value from the activity monitor rises above this threshold, DPM transitions to the next higher performance state. |

### GMMx7[6C:50] SCLK DPM Activity Sampling Parameters 7-0

Reset: 0000_0000h. See 2.5.5.1.1 [Activity Monitor] and 2.5.5.1.2 [SCLK DPM].

**Table 134: DPM state mapping for** GMMx7[6C:50]

| Register | Function | Register | Function |
|---|---|---|---|
| GMMx750 | State 0 | GMMx760 | State 4 |
| GMMx754 | State 1 | GMMx764 | State 5 |
| GMMx758 | State 2 | GMMx768 | State 6 |
| GMMx75C | State 3 | GMMx76C | State 7 |

| Bits | Description |
|---|---|
| 31:16 | **BusySamplingUnit**. Read-write. See GMMx7[6C:50][BusySamplingPeriod]. |
| 15:0 | **BusySamplingPeriod**. Read-write. Specifies the period at which SCLK DPM samples activity in REFCLKs. Period = (4^BusySamplingUnit) * BusySamplingPeriod. |

### GMMx770 CG Voltage Control

Reset: 0000_0006h. See 2.5.1.5.2 [Software-Initiated Voltage Transitions].

| Bits | Description |
|---|---|
| 31:5 | Reserved. |
| 4 | **VoltageForceEn**. Read-write. See D0F0x64_x6A[VoltageForceEn]. |
| 3 | **VoltageChangeEn**. Read-write. See D0F0x64_x6A[VoltageChangeEn]. |
| 2:1 | **VoltageLevel**. Read-write. See D0F0x64_x6A[VoltageLevel]. |
| 0 | **VoltageChangeReq**. Read-write. See D0F0x64_x6A[VoltageChangeReq]. |

### GMMx774 CG Voltage Status

Reset: 0000_0000h. See 2.5.1.5.2 [Software-Initiated Voltage Transitions].

| Bits | Description |
|---|---|
| 31:3 | Reserved. |
| 2:1 | **CurrentVoltageLevel**. Read-only. See D0F0x64_x6B[CurrentVoltageLevel]. |
| 0 | **VoltageChangeAck**. Read-only. See D0F0x64_x6B[VoltageChangeAck]. |

### GMMx788 DPM Voltage Control

Reset: 0000_FFFFh. See 2.5.5.1.2 [SCLK DPM].

| Bits | Description |
|---|---|
| 31:17 | Reserved. |
| 16 | **DpmVidChangeEn**. Read-write. 1=VID changes are requested as a result of DPM state transitions. 0=DPM state transitions do not cause VID change requests. |
| 15:14 | **Dpm7Vid**. Read-write. See GMMx788[Dpm0Vid]. |
| 13:12 | **Dpm6Vid**. Read-write. See GMMx788[Dpm0Vid]. |
| 11:10 | **Dpm5Vid**. Read-write. See GMMx788[Dpm0Vid]. |
| 9:8 | **Dpm4Vid**. Read-write. See GMMx788[Dpm0Vid]. |
| 7:6 | **Dpm3Vid**. Read-write. See GMMx788[Dpm0Vid]. |
| 5:4 | **Dpm2Vid**. Read-write. See GMMx788[Dpm0Vid]. |
| 3:2 | **Dpm1Vid**. Read-write. See GMMx788[Dpm0Vid]. |
| 1:0 | **Dpm0Vid**. Read-write. Specifies the VID requested when the processor enters the respective DPM state. This field indexes into D18F3x15C as follows:<br>Bits — VID code<br>00b — D18F3x15C[SclkVidLevel0]<br>01b — D18F3x15C[SclkVidLevel1]<br>10b — D18F3x15C[SclkVidLevel2]<br>11b — D18F3x15C[SclkVidLevel3] |

### GMMx830 SCLK DPM Control 10

Reset: 00B6_DB6Dh. See 2.5.5.2 [LCLK Clock Ramping (Deep Sleep)].

**Table 135: SCLK shallow sleep to DPM state mapping**

| Register | Bits | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 23:21 | 20:18 | 17:15 | 14:12 | 11:9 | 8:6 | 5:3 | 2:0 |
| GMMx830 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

| Bits | Description |
|---|---|
| 31:24 | Reserved. |
| 23:21 | **FstateSsDivId: DPM state shallow sleep divisor**. Read-write. See GMMx830[2:0]. |
| 20:18 | **FstateSsDivId: DPM state shallow sleep divisor**. Read-write. See GMMx830[2:0]. |
| 17:15 | **FstateSsDivId: DPM state shallow sleep divisor**. Read-write. See GMMx830[2:0]. |
| 14:12 | **FstateSsDivId: DPM state shallow sleep divisor**. Read-write. See GMMx830[2:0]. |
| 11:9 | **FstateSsDivId: DPM state shallow sleep divisor**. Read-write. See GMMx830[2:0]. |
| 8:6 | **FstateSsDivId: DPM state shallow sleep divisor**. Read-write. See GMMx830[2:0]. |

| | |
|---|---|
| 5:3 | **FstateSsDivId: DPM state shallow sleep divisor**. Read-write. See GMMx830[2:0]. |
| 2:0 | **FstateSsDivId: DPM state shallow sleep divisor**. Read-write. Specifies the shallow sleep divisor used when in the respective DPM state.<br>Bits Divisor Bits Divisor<br>0h Clock off. 4h Divide by 16.<br>1h Divide by 2. 5h Divide by 32.<br>2h Divide by 4. 7h-6h Reserved.<br>3h Divide by 8. |

### 3.17.2 Graphics Memory Controller (GMC) Registers

#### GMMx15C0 VM L2 Domain Clock Gate Control

See 2.5.5.3 [GPU and Root Complex Clock Gating].

| Bits | Description |
|---|---|
| 31:19 | Reserved. |
| 18 | **Enable**. Read-write. Reset: 1. BIOS: 2.14.2.1. 0=Always on SCLK. 1=Use clock gating. |
| 17:12 | **RdyDly**. Read-write. Reset: 1. Delay in clock cycles to turn on rdy after the clock starts. |
| 11:6 | **OffDly**. Read-write. Reset: 10h. Delay in clock cycles to turn off clock after busy goes low. |
| 5:0 | **OnDly**. Read-write. Reset: 1. Delay in clock cycles to turn on the clock after requested. |

#### GMMx2014 GMC Read Group Weight 0

BIOS: 0000_5500h. Specifies the weight the client uses for read group arbitration.

| Bits | Description |
|---|---|
| 31:16 | Reserved. |
| 15:12 | **Mcif**. Read-write. Reset: 3h. |
| 11:8 | **Dmif**. Read-write. Reset: 3h. |
| 7:4 | **Vmc**. Read-write. Reset: 0. |
| 3:0 | **Rlc**. Read-write. Reset: 0. |

#### GMMx2018 GMC Write Group System

BIOS: 0000_0050h. Specifies the weight the client uses for write group arbitration.

| Bits | Description |
|---|---|
| 31:16 | Reserved. |
| 15:12 | **Vip**. Read-write. Reset: 0. |
| 11:8 | **Rlc**. Read-write. Reset: 0. |
| 7:4 | **Mcif**. Read-write. Reset: 0. |
| 3:0 | **Ih**. Read-write. Reset: 0. |

### GMMx201C GMC Read Group Weight 2

BIOS: 0333_0003h. Specifies the weight the client uses for read group arbitration.

| Bits | Description |
|---|---|
| 31:28 | Reserved. |
| 27:24 | **UvdExt1**. Read-write. Reset: 0. |
| 23:20 | **Uvd**. Read-write. Reset: 0. |
| 19:16 | **Umc**. Read-write. Reset: 0. |
| 15:12 | **Sem**. Read-write. Reset: 0. |
| 11:8 | **Hdp**. Read-write. Reset: 0. |
| 7:4 | **DrmDma**. Read-write. Reset: 0. |
| 3:0 | **UvdExt0**. Read-write. Reset: 0. |

### GMMx2020 GMC Write Group Weight

BIOS: 7076_0007h. Specifies the weight the client uses for write group arbitration.

| Bits | Description |
|---|---|
| 31:28 | **UvdExt1**. Read-write. Reset: 0. |
| 27:24 | **Xdp**. Read-write. Reset: 0. |
| 23:20 | **Uvd**. Read-write. Reset: 0. |
| 19:16 | **Umc**. Read-write. Reset: 0. |
| 15:12 | **Sem**. Read-write. Reset: 0. |
| 11:8 | **Hdp**. Read-write. Reset: 0. |
| 7:4 | **DrmDma**. Read-write. Reset: 0. |
| 3:0 | **UvdExt0**. Read-write. Reset: 0. |

### GMMx2024 GMC Frame Buffer Location

Reset: 0000_0000h. The register along with [The GMC Frame Buffer Offset] GMMx2898 specifies the base address of the frame buffer in GPU physical memory and system memory. See 2.14.3 [Frame Buffer (FB)].

| Bits | Description |
|---|---|
| 31:16 | **Top: frame buffer region GPU limit address[39:24]**. Read-write. Specifies the GPU address of the top of frame buffer region 0. |
| 15:0 | **Base: frame buffer GPU base address[30:24]**. Read-write. Specifies the GPU base address of the frame buffer. |

### GMMx2028 GMC VM Noncoherent System Memory Top

BIOS: 0000_0000h.

| Bits | Description |
|---|---|
| 31:18 | Reserved. |
| 17:0 | **SysTop[39:22]**. Read-write. Reset: 0. |

### GMMx202C GMC VM Noncoherent System Memory Bottom

BIOS: 0003_FFFFh.

| Bits | Description |
|---|---|
| 31:18 | Reserved. |
| 17:0 | **SysBot[39:22]**. Read-write. Reset: 0. |

### GMMx20[C0:B8] GMC Miscellaneous Domain Clock Gate Control

See 2.5.5.3 [GPU and Root Complex Clock Gating].

| Bits | Description |
|---|---|
| 31:19 | Reserved. |
| 18 | **Enable**. Read-write. Reset: 1. BIOS: 2.14.2.1. 1=Clock gating enabled. |
| 17:12 | **RdyDly**. Read-write. Reset: 0. Delay in clock cycles (RdyDly+1) to turn on rdy after the clock starts. |
| 11:6 | **OffDly**. Read-write. IF (REG==GMMx20C0) THEN Reset: 32h. ELSE Reset: 10h. ENDIF. Delay in clock cycles (OffDly+1) to turn off clock after busy goes low. |
| 5:0 | **OnDly**. Read-write. Reset: 0. Delay in clock cycles (OnDly+1) to turn on the clock after requested. |

### GMMx20D4 GMC Wide Data Path Control

| Bits | Description |
|---|---|
| 31:1 | Reserved. |
| 0 | **LocalBlackout**. Read-write. Reset: 1. 1=Prevent all non-exempt clients from receiving a go. |

### GMMx20EC GMC Read Request Control

| Bits | Description |
|---|---|
| 31:2 | Reserved. |
| 1 | **LocalBlackout**. Read-write. Reset: 1. 1=Prevents all non-exempt clients from receiving a go. |
| 0 | **RemoteBlackout**. Read-write. Reset: 1. 1=Prevents all non-exempt mcd requests from getting a go. |

### GMMx2114 GMC Read Request Credits

BIOS: 0000_000Ch.

| Bits | Description |
|---|---|

| | |
|---|---|
| 31:8 | Reserved. |
| 7:0 | **Stor1Pri**. Read-write. Reset: 10h. Total depth of stor1 for priority. |

### GMMx2144 GMC Read Request Control

| Bits | Description |
|---|---|
| 31:18 | Reserved. |
| 17:11 | **AskCredits**. Read-write. Reset: 20h. BIOS: 18h. Specifies the number of arbitration credits available. |
| 10:0 | Reserved. |

### GMMx21[8C:60] GMC Read Request Client Control

IF (REG==GMMx217C) THEN BIOS: 0000_A1F1.
ELSEIF (REG==GMMx2188) THEN BIOS: 0002_21B1h. ENDIF.

| Bits | Description |
|---|---|
| 31:20 | Reserved. |
| 19:16 | IF (REG==GMMx2188) THEN **ReqLimit**. Read-write. Reset: Fh. Read requests are halted when the number of outstanding reads exceeds ReqLimit*32 and GMC is not in stutter mode. Fh=No limit. ELSE Reserved. ENDIF. |
| 15 | **StallOverrideWtm**. Read-write. Reset: 0. 1=Stall signal is set to the or of the bits in StallMode. |
| 14:11 | **LazyTimer**. Read-write. Reset: 4h. Maximum number of cycles to wait after 1st ask before the first go can be issued. |
| 10:7 | **MaxBurst**. Read-write. Reset: 3h. Maximum number of gos (MaxBurst+1) given to a client before arbitration changes to another client. |
| 6 | **StallOverride**. Read-write. Reset: 0. 1=Stall signal is set to the or of the bits in StallMode. |
| 5:4 | **StallMode**. Read-write. Reset: 1. How the client stall signal is used in weight manager.<br>Bits Definition Bits Definition<br>0h Ignored. 2h Quadruple arbitration weight.<br>1h Double arbitration weight. 3h Maximum value to arbitration weight. |
| 3 | **BlackoutExempt**. Read-write. Reset: 0. 1=Client is exempt from GMMx20EC[LocalBlackout] and can still make memory requests. |
| 2:1 | **Prescale**. Read-write. Reset: 0. Prescaler for client urgency signal and outstanding request count.<br>Bits Definition Bits Definition<br>0h Not changed. 2h Prescaled by 2.<br>1h Prescaled by 1/2. 3h Reserved. |
| 0 | **Enable**. Read-write. Reset: 1. 1=Enable client. |

### GMMx219[C:0] GMC Wide Data Path Control

| Bits | Description |
|---|---|
| 31:23 | Reserved. |
| 22:17 | **StallThreshold**. Read-write. Reset: 20h. Threshold for when hub should assert the stall signal to mcd. |

| | |
|---|---|
| 16:13 | **LazyTimer**. Read-write. Reset: 4h. Maximum number of cycles to wait after first ask before the first go can be issued. |
| 12:7 | **AskCredits**. Read-write. Reset: 20h. Credits from internal hub arbitration to internal ask FIFO. |
| 6:3 | **MaxBurst**. Read-write. Reset: 3h. Maximum number of gos (MaxBurst+1) given to a wrreq sourced from this client before arbitration changes to another client. |
| 2 | **StallMode**. Read-write. Reset: 1. 1=Use stall signal. |
| 1 | **BlackoutExempt**. Read-write. Reset: 0. 1=Client is exempt from and can still make memory requests. |
| 0 | **Enable**. Read-write. Reset: 1. 1=Enable client. |

### GMMx21[D0:A4] GMC Wide Data Path Client Control

BIOS: IF (REG==GMMx21C8) THEN 0000_A1F1h. ENDIF.

| Bits | Description |
|---|---|
| 31:16 | Reserved. |
| 15 | **StallOverrideWtm**. Read-write. Reset: 0. 1=Stall signal is set to the or of the bits in StallMode. |
| 14:11 | **LazyTimer**. Read-write. Reset: 4h. Maximum number of cycles to wait after the first ask before the first go can be issued. |
| 10:7 | **MaxBurst**. Read-write. Reset: 3h. Maximum number of gos (MaxBurst+1) given to a client before arbitration changes to another client. |
| 6 | **StallOverride**. Read-write. Reset: 0. 1=Stall signal is set to the or of the bits in StallMode. |
| 5:4 | **StallMode**. Read-write. Reset: 1. How the client stall signal is used in weight manager.<br>Bits Definition Bits Definition<br>0h Ignored. 2h Quadruple arbitration weight.<br>1h Double arbitration weight. 3h Maximum value to arbitration weight. |
| 3 | **BlackoutExempt**. Read-write. Reset: 0. 1=exempt from [The GMC Wide Data Path Control] GMMx20D4[LocalBlackout], can still make memory requests. |
| 2:1 | **Prescale**. Read-write. Reset: 0. Prescaler for client urgency signal and outstanding request count.<br>Bits Definition Bits Definition<br>0h Not changed. 2h Prescaled by 2.<br>1h Prescaled by 1/2. 3h Reserved. |
| 0 | **Enable**. Read-write. Reset: 1. 1=Enable client. |

### GMMx2478 GMC Clock Gating Control

See 2.5.5.3 [GPU and Root Complex Clock Gating].

| Bits | Description |
|---|---|
| 31:19 | Reserved. |
| 18 | **Enable**. Read-write. Reset: 1. BIOS: 2.14.2.1. 1=Clock gating enabled. |
| 17:12 | **RdyDly**. Read-write. Reset: 0. Delay in clock cycles (RdyDly+1) to turn on rdy after the clock starts. |
| 11:6 | **OffDly**. Read-write. Reset: 10h. Delay in clock cycles to turn off clock after busy goes low. |
| 5:0 | **OnDly**. Read-write. Reset: 0. Delay in clock cycles (OnDly+1) to turn on the clock after requested. |

**GMMx25C0 GMC Master Client Interface Control**

| Bits | Description |
|---|---|
| 31:2 | Reserved. |
| 1 | **BlackoutWr**. Read-write. Reset: 1. |
| 0 | **BlackoutRd**. Read-write. Reset: 1. |

**GMMx25C8 Read Interface Depth**

BIOS: 007F_605F.

| Bits | Description |
|---|---|
| 31:26 | Reserved. |
| 25 | **HubPri**. Read-write. Reset: 0. |
| 24 | **LclPri**. Read-write. Reset: 0. |
| 23:16 | **ReadPri**. Read-write. Reset: 7Fh. Specifies the number of entries available for high priority remote read requests.<br>Bits Definition<br>FFh-00h <ReadPri*2> entries |
| 15:8 | **ReadHub**. Read-write. Reset: 60h. Specifies the number of entries available for low priority remote read requests.<br>Bits Definition<br>FFh-00h <ReadHub*2> entries |
| 7:0 | **ReadLcl**. Read-write. Reset: 60h. Specifies the number of entries available for local read requests.<br>Bits Definition<br>FFh-00h <ReadLcl*2> entries |

**GMMx25CC GMC Write Interface Depth**

BIOS: 0000_7F7Eh.

| Bits | Description |
|---|---|
| 31:17 | Reserved. |
| 16 | **HubPri**. Read-write. Reset: 0. |
| 15:8 | **WriteHub**. Read-write. Reset: 7Fh. Specifies the number of entries available for remote write requests.<br>Bits Definition<br>FFh-00h <WriteHub*2> entries |
| 7:0 | **WriteLcl**. Read-write. Reset: 7Fh. Specifies the number of entries available for local write requests.<br>Bits Definition<br>FFh-00h <WriteLcl*2> entries |

**GMMx2620 GMC Local Read Group Weight 0**

BIOS: 4411_1222h. Specifies the weight the local client uses for read group arbitration.

| Bits | Description |
|---|---|
| 31:28 | **DbhTile0**. Read-write. Reset: 1. |
| 27:24 | **Db0**. Read-write. Reset: 1. |
| 23:20 | **CbfMask0**. Read-write. Reset: 1. |
| 19:16 | **CbcMask0**. Read-write. Reset: 1. |
| 15:12 | **Cb0**. Read-write. Reset: 1. |
| 11:8 | **Vc0**. Read-write. Reset: 2h. |
| 7:4 | **TcvFetch0**. Read-write. Reset: 2h. |
| 3:0 | **TctFetch0**. Read-write. Reset: 2h. |

### GMMx2624 GMC Local Write Group Weight 0

BIOS: 1133_3111h. Specifies the weight the local client uses for write group arbitration.

| Bits | Description |
|---|---|
| 31:28 | **Cbimmed0**. Read-write. Reset: 1. |
| 27:24 | **Bcast0**. Read-write. Reset: 1. |
| 23:20 | **Sx0**. Read-write. Reset: 1. |
| 19:16 | **DbhTile0**. Read-write. Reset: 1. |
| 15:12 | **Db0**. Read-write. Reset: 1. |
| 11:8 | **CbfMask0**. Read-write. Reset: 1. |
| 7:4 | **CbcMask0**. Read-write. Reset: 1. |
| 3:0 | **Cb0**. Read-write. Reset: 1. |

### GMMx2628 GMC Local Read Group Weight 1

BIOS: 4411_1666h. Specifies the weight the local client uses for read group arbitration.

| Bits | Description |
|---|---|
| 31:28 | **DbhTile1**. Read-write. Reset: 1. |
| 27:24 | **Db1**. Read-write. Reset: 1. |
| 23:20 | **CbfMask1**. Read-write. Reset: 1. |
| 19:16 | **CbcMask1**. Read-write. Reset: 1. |
| 15:12 | **Cb1**. Read-write. Reset: 1. |
| 11:8 | **Vc1**. Read-write. Reset: 6h. |
| 7:4 | **TcvFetch1**. Read-write. Reset: 6h. |
| 3:0 | **TctFetch1**. Read-write. Reset: 6h. |

### GMMx262C GMC Local Write Group Weight 1

BIOS: 2144_4222h. Specifies the weight the local client uses for write group arbitration.

| Bits | Description |
|---|---|
| 31:28 | **Cbimmed1**. Read-write. Reset: 1. |
| 27:24 | **Bcast1**. Read-write. Reset: 1. |
| 23:20 | **Sx1**. Read-write. Reset: 1. |
| 19:16 | **DbhTile1**. Read-write. Reset: 1. |
| 15:12 | **Db1**. Read-write. Reset: 1. |
| 11:8 | **CbfMask1**. Read-write. Reset: 1. |
| 7:4 | **CbcMask1**. Read-write. Reset: 1. |
| 3:0 | **Cb1**. Read-write. Reset: 1. |

### GMMx2630 GMC External Read Group Weight

BIOS: 0000_0044h. Specifies the weight the external client uses for read group arbitration.

| Bits | Description |
|---|---|
| 31:8 | Reserved. |
| 7:4 | **DbstEn1**. Read-write. Reset: 1. |
| 3:0 | **DbstEn0**. Read-write. Reset: 1. |

### GMMx2634 GMC External Write Group Weight

BIOS: 0000_0043h. Specifies the weight the external client uses for write group arbitration.

| Bits | Description |
|---|---|
| 31:8 | Reserved. |
| 7:4 | **DbstEn1**. Read-write. Reset: 1. |
| 3:0 | **DbstEn0**. Read-write. Reset: 1. |

### GMMx26[58:50] GMC Clock Gating Control

See 2.5.5.3 [GPU and Root Complex Clock Gating].

| Bits | Description |
|---|---|
| 31:19 | Reserved. |
| 18 | **Enable**. Read-write. Reset: 1. BIOS: 2.14.2.1. 1=Clock gating enabled. |
| 17:12 | **RdyDly**. Read-write. Reset: 0. delay in clock cycles (+1) to turn on rdy after the clock starts. |
| 11:6 | **OffDly**. Read-write. Reset: 10h. delay in clock cycles (+1) to turn off the clock after busy goes low. |
| 5:0 | **OnDly**. Read-write. Reset: 0. delay in clock cycles (+1) to turn on the clock after requested. |

### GMMx2[8D8,77C] GMC DRAM Timing

Reset: 1919_0A0Ah.

| Bits | Description |
|---|---|
| 31:24 | **RasMActWr: Trc - Trcdw**. Read-write. BIOS: MIN((D18F2x[1,0]F4_x40[Trc]+11)-(D18F2x[1,0]F4_x40[Trcd]+5)). |
| 23:16 | **RasMActRd: Trc - Trcdr**. Read-write. BIOS: MIN((D18F2x[1,0]F4_x40[Trc]+11)-(D18F2x[1,0]F4_x40[Trcd]+5)). |
| 15:8 | **ActWr: Trcdw**. Read-write. BIOS: MIN(D18F2x[1,0]F4_x40[Trcd]) + 5. |
| 7:0 | **ActRd: Trcdr**. Read-write. BIOS: MIN(D18F2x[1,0]F4_x40[Trcd]) + 5. |

### GMMx2[8DC,780] GMC DRAM Timing 2

Reset: 0D14_0A23h.

| Bits | Description |
|---|---|
| 31:24 | **BusTurn: bus turnaround time**. Read-write. BIOS: MIN(D18F2x[1,0]84[Tcwl] + D18F2x[1,0]F4_x41[Twtr] + D18F2x[1,0]8C[TrwtTO]) + 15)/2. |
| 23:16 | **WrPlusRp: Twr + Trp**. Read-write.<br>IF (MIN(D18F2x084[Twr], D18F2x184[Twr]) < 100b) THEN<br>BIOS: MIN(D18F2x[1,0]F4_x40[Trp]+D18F2x[1,0]84[Twr]+8).<br>ELSE<br>BIOS: MIN((D18F2x[1,0]F4_x40[Trp]+4)+(D18F2x[1,0]84[Twr]*2)).<br>ENDIF. |
| 15:8 | **Rp: Trp - 1**. Read-write. BIOS: MIN(D18F2x[1,0]F4_x40[Trp]) + 4. |
| 7:0 | **Ras2Ras: Trc -1**. Read-write. BIOS: MIN(D18F2x[1,0]F4_x40[Trc]) + 10. |

### GMMx27[88:84] Weight Manager Control

BIOS: IF (REG== GMMx2784) THEN 0000_0007h. ENDIF.

| Bits | Description |
|---|---|
| 31:3 | Reserved. |
| 2 | **HarshPri**. Read-write. Reset: 0. 1=If any priority groups have valid requests, then all other groups weights are invalid. |
| 1:0 | **WtMode**. Read-write. Reset: 3h. |

### GMMx27[A0:9C] GMC Group 3-0 Read and Write Arbitration Timer Control

BIOS: FCFC_FDFCh.

| Bits | Description |
|---|---|
| 31:24 | **Group3**. Read-write. Reset: FCh. |
| 23:16 | **Group2**. Read-write. Reset: FCh. |
| 15:8 | **Group1**. Read-write. Reset: FCh. |
| 7:0 | **Group0**. Read-write. Reset: FCh. |

### GMMx27[D0:CC] List Manager Control

BIOS: IF (REG== GMMx27CC) THEN 0003_2005h. ELSE 0001_2008h. ENDIF.

| Bits | Description |
|---|---|
| 31:18 | Reserved. |
| 17 | **StreakUber**. Read-write. Reset: 1. |
| 16 | **StreakBreak**. Read-write. Reset: 1. |
| 15:8 | **StreakLimitUber**. Read-write. Reset: 20h. number of read transactions allowed to the same page (a streak) before forcing a page close when streak_uber is set and the client group is rdy. |
| 7:0 | **StreakLimit**. Read-write. Reset: 8h. number of read transactions allowed to the same page (a streak) before forcing a page close programmed value = max_streak_length-2*8. |

### GMMx27DC GMC Read Arbitration Credit Control

BIOS: 0073_4847h.

| Bits | Description |
|---|---|
| 31:24 | Reserved. |
| 23:16 | **Disp**. Read-write. Reset: 60h. Specifies the number of credits allocated for display reads. |
| 15:8 | **Hub**. Read-write. Reset: 40h. Specifies the number of credits allocated for remote client reads. |
| 7:0 | **Lcl**. Read-write. Reset: 3Fh. Specifies the number of credits allocated for local client reads. |

### GMMx27E0 GMC Write Arbitration Credit Control

BIOS: 0000_3D3Ch.

| Bits | Description |
|---|---|
| 31:16 | Reserved. |
| 15:8 | **Hub**. Read-write. Reset: 40h. Specifies the number of credits allocated for remote client writes. |
| 7:0 | **Lcl**. Read-write. Reset: 3Fh. Specifies the number of credits allocated for local client writes |

### GMMx2814 UVD Read Latency Control

BIOS: 0000_0220h.

| Bits | Description |
|---|---|
| 31:10 | Reserved. |
| 9 | **UvdHarshPriority**. Read-write. Reset: 0. BIOS: 1. 1=Urgent UVD reads are only number 2 priority only behind urgent display reads. |
| 8:0 | **WriteClks**. Read-write. Reset: 20h. BIOS: 0. |

### GMMx28[38:1C] GMC DCT CS Base Address

The information in these registers is copied from [The DRAM CS Base Address Registers]

D18F2x[1,0][4C:40].

**Table 136: GMC CS base address register mapping and reset values**

| Register | Function | Reset | DCT Register |
|---|---|---|---|
| GMMx281C | DCT0 CS0 | 0000_0001h | D18F2x40 |
| GMMx2820 | DCT1 CS0 | 0000_0001h | D18F2x140 |
| GMMx2824 | DCT0 CS1 | 0000_0081h | D18F2x44 |
| GMMx2828 | DCT1 CS1 | 0000_0081h | D18F2x144 |
| GMMx282C | DCT0 CS2 | 0000_0101h | D18F2x48 |
| GMMx2830 | DCT1 CS2 | 0000_0101h | D18F2x148 |
| GMMx2834 | DCT0 CS3 | 0000_0181h | D18F2x4C |
| GMMx2838 | DCT1 CS3 | 0000_0181h | D18F2x14C |

| Bits | Description |
|---|---|
| 31:29 | Reserved. |
| 28:19 | **BaseAddr[36:27]: normalized physical base address bits [36:27]**. Read-write. |
| 18:14 | Reserved. |
| 13:5 | **BaseAddr[21:13]: normalized physical base address bits [21:13]**. Read-write. |
| 4:1 | Reserved. |
| 0 | **CSEnable: chip select enable**. Read-write. |

### GMMx28[48:3C] GMC DCT[1:0] CS[3:0] Mask

Reset: 01F8_3CE0h. The information in these registers is copied from [The DRAM CS Mask Register] D18F2x[1,0][64:60].

**Table 137: GMC CS mask register mapping**

| Register | Function | DCT Register |
|---|---|---|
| GMMx283C | DCT0 CS[1:0] | D18F2x60 |
| GMMx2840 | DCT0 CS[3:2] | D18F2x64 |
| GMMx2844 | DCT1 CS[1:0] | D18F2x160 |
| GMMx2848 | DCT1 CS[3:2] | D18F2x164 |

| Bits | Description |
|---|---|
| 31:29 | Reserved. |
| 28:19 | **AddrMask[36:27]: normalized physical address mask bits [36:27]**. Read-write. |
| 18:14 | Reserved. |
| 13:5 | **AddrMask[21:13]: normalized physical address mask bits [21:13]**. Read-write. |
| 4:0 | Reserved. |

### GMMx28[50:4C] GMC DCT[1:0] Bank Address Mapping

Reset: 0002_0077h.

**Table 138: GMC bank address register mapping**

| Register | Function | DCT Register |
|---|---|---|
| GMMx284C | DCT 0 | D18F2x80 |
| GMMx2850 | DCT 1 | D18F2x180 |

| Bits | Description |
|---|---|
| 31:20 | Reserved. |
| 19 | **BankSwap: bank swap**. Read-write. BIOS: D18F2x[1,0]A8[BankSwap]. |
| 18 | **BurstLength32: burst length**. Read-write. 1=32-byte mode. 0=64-byte mode. |
| 17 | **Ddr3Mode: DDR3 mode**. Read-write. 1=DDR3. |
| 16 | **BankSwizzleMode: bank swizzle mode**. Read-write. BIOS: D18F2x[1,0]94[BankSwizzleMode]. |
| 15:8 | Reserved. |
| 7:4 | **Dimm1AddrMap: DIMM 1 address map**. Read-write. BIOS: D18F2x[1,0]80[Dimm1AddrMap]. |
| 3:0 | **Dimm0AddrMap: DIMM 0 address map**. Read-write. BIOS: D18F2x[1,0]80[Dimm0AddrMap]. |

### GMMx2854 GMC DRAM Control

Reset: 0002_0004h.

| Bits | Description |
|---|---|
| 31:24 | Reserved. |
| 23:11 | **DctSelBaseAddr[39:27]: DRAM controller select base address bits[39:27]**. Read-write. BIOS: D18F2x110[DctSelBaseAddr]. |
| 10:8 | Reserved. |
| 7:6 | **DctSelIntLvAddr[1:0]: DRAM controller select channel interleave address bit**. Read-write. BIOS: D18F2x110[DctSelIntLvAddr]. |
| 5:3 | Reserved. |
| 2 | **DctSelIntLvEn: DRAM controller interleave enable**. Read-write. BIOS: D18F2x110[DctSelIntLvEn]. |
| 1 | **DctSelHi: DRAM controller high select**. Read-write. BIOS: D18F2x110[DctSelHi]. |
| 0 | **DctSelHiRngEn: DRAM controller select high range enable**. Read-write. BIOS: D18F2x110[DctSelHiRngEn]. |

### GMMx2858 GMC DRAM Control 2

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:24 | Reserved. |
| 23:10 | **DctSelBaseOffset[39:26]: DRAM controller select base offset address bits[39:26]**. Read-write. BIOS: D18F2x114[DctSelBaseOffset]. |

| | |
|---|---|
| 9 | **DctSelIntLvAddr[2]: DRAM controller select channel interleave address bit**. Read-write. BIOS: D18F2x114[DctSelIntLvAddr]. |
| 8:0 | Reserved. |

### GMMx285C GMC DRAM Hole Address

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:24 | **DramHoleBase[31:24]: DRAM hole base address**. Read-write. BIOS: D18F1xF0[DramHoleBase]. |
| 23:16 | Reserved. |
| 15:7 | **DramHoleOffset[31:23]: DRAM hole offset address**. Read-write. BIOS: D18F1xF0[DramHoleOffset]. |
| 6:1 | Reserved. |
| 0 | **DramHoleValid**. Read-write. BIOS: D18F1xF0[DramHoleValid]. |

### GMMx2860 GMC DRAM Interleave Region

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:16 | Reserved. |
| 15:11 | **IntLvRegionLimit: interleaved region limit address**. Read-write. BIOS: D18F2x10C[IntLvRegionLimit]. |
| 10:8 | Reserved. |
| 7:3 | **IntLvRegionBase: interleaved region base address**. Read-write. BIOS: D18F2x10C[IntLvRegionBase]. |
| 2:1 | Reserved. |
| 0 | **IntLvRegionEn: interleaved region remap enable**. Read-write. BIOS: D18F2x10C[IntLvRegionEn]. |

### GMMx2864 DRAM Address Swizzle0

IF (DualChannel) THEN BIOS: 3200_9817h. ELSE BIOS: 3210_0876h. ENDIF. This register is used to swizzle address bits for frame buffer accesses only.

| Bits | Description |
|---|---|
| 31:28 | **A15Map**. Read-write. Reset: 0. |
| 27:24 | **A14Map**. Read-write. Reset: 0. |
| 23:20 | **A13Map**. Read-write. Reset: 0. |
| 19:16 | **A12Map**. Read-write. Reset: 0. |
| 15:12 | **A11Map**. Read-write. Reset: 0. |
| 11:8 | **A10Map**. Read-write. Reset: 0. |

| | |
|---|---|
| 7:4 | **A9Map**. Read-write. Reset: 0. |
| 3:0 | **A8Map**. Read-write. Reset: 0. |

### GMMx2868 DRAM Address Swizzle 1

IF (DualChannel) THEN BIOS: 0000_0004h. ELSE BIOS: 0000_0000h. ENDIF. This register is used to swizzle address bits for frame buffer accesses only.

| Bits | Description |
|---|---|
| 31:16 | Reserved. |
| 15:12 | **A19Map**. Read-write. Reset: 0. |
| 11:8 | **A18Map**. Read-write. Reset: 0. |
| 7:4 | **A17Map**. Read-write. Reset: 0. |
| 3:0 | **A16Map**. Read-write. Reset: 0. |

### GMMx28[78:6C] GMC DRAM Aperture Base 3-0

Reset: 0000_0000h.

The GMC DRAM Aperture Base/Limit registers specify the portions of the frame buffer that are accessible when D18F3xF8_x4[SecureGfxMode]=1. All frame buffer accesses that do not match one of the apertures are forwarded to the 4KByte region of the frame buffer specified by [The GMC DRAM Aperture Default Base Address] GMMx2894.

| Bits | Description |
|---|---|
| 31:20 | Reserved. |
| 19:0 | **Base: frame buffer aperture base address[39:20]**. Read-write. |

### GMMx28[88:7C] GMC DRAM Aperture Limit 3-0

Reset: 000F_FFFFh.

| Bits | Description |
|---|---|
| 31:20 | Reserved. |
| 19:0 | **Top: frame buffer aperture limit address[39:20]**. Read-write. |

### GMMx288C GMC C6 Save Base

Reset: 0000_0000h. The GMC C6 Save Base/Limit registers specify the portion of DRAM used for saving C6 data. All accesses that match the save region are forwarded to the 4KByte region of the frame buffer specified by [The GMC DRAM Aperture Default Base Address] GMMx2894.

| Bits | Description |
|---|---|
| 31:20 | Reserved. |
| 19:0 | **Base: C6 save base address[39:20]**. Read-write. BIOS:{D18F4x12C[C6Base],0h} |

### GMMx2890 GMC C6 Save Limit

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:20 | Reserved. |
| 19:0 | **Top: C6 save limit address[39:20]**. Read-write. BIOS:{D18F4x12C[C6Base],Fh} |

### GMMx2894 GMC DRAM Aperture Default Base Address

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:28 | Reserved. |
| 27:0 | **Def: frame buffer aperture default base address[39:12]**. Read-write. |

### GMMx2898 GMC Frame Buffer Offset

Reset: 0F00_0000h.

The register along with specifies the base address of the frame buffer in GPU physical memory and system memory. See 2.14.3 [Frame Buffer (FB)].

| Bits | Description |
|---|---|
| 31:28 | Reserved. |
| 27:24 | **Top: frame buffer region GPU limit address[23:20]**. Read-write. Specifies the GPU address of the top of frame buffer region. |
| 23:20 | **Base: frame buffer GPU base address[23:20]**. Read-write. Specifies the GPU base address of the frame buffer. |
| 19:0 | **Offset: frame buffer region system base address[39:20]**. Read-write. Specifies the system base address of frame buffer region. |

### GMMx28C8 GMC IFQ Arbitration

Reset: 0000_0004h.

| Bits | Description |
|---|---|
| 31:4 | Reserved. |
| 3:0 | **Delay: IFQ check delay**. Read-write. BIOS: 3h. This field specifies the number of clocks to wait for the IFQ check response. |

### GMMx28EC GMC DCT Interface Control

BIOS: 0018_7000h.

| Bits | Description |
|---|---|
| 31:22 | Reserved. |

| | |
|---|---|
| 21:18 | **DctCredits**. Read-write. Reset: 4h. Specifies the number of credits on the internal DCT interface. |
| 17:0 | Reserved. |

### GMMx2B8C GMC Register Engine RAM Index

This register along GMMx2B90 is used to access the register engine RAM.

| Bits | Description |
|---|---|
| 31:10 | Reserved. |
| 9:0 | **RengRamIndex**. Read-write. Reset: 0. Index into RENG RAM. This field increments after GMMx2B90 is read or written. |

### GMMx2B90 GMC Register Engine RAM Data

This register along GMMx2B8C is used to access the register engine RAM.

| Bits | Description |
|---|---|
| 31:0 | **RengRamData**. Read-write. Reset: 0. Data pointed to by GMMx2B90[RengRamIndex] |

### GMMx2B94 GMC RENG Execution Control

| Bits | Description |
|---|---|
| 31:1 | Reserved. |
| 0 | **RengExecuteOnPwrUp**. Read-write. Reset: 0. 1=The RENG executes when GMC resumes from a power gated state. |

### GMMx2B98 GMC Miscellaneous Controls

| Bits | Description |
|---|---|
| 31:28 | Reserved. |
| 27 | **CriticalRegsLock**. Read-write. Reset: 0. BIOS: 1. 1=Critical registers in the GMC and RengExecuteNonsecureStartPtr re read only. |
| 26:17 | Reserved. |
| 16 | **StctrlStutterEn**. Read-write. Reset: 0. Enables stutter mode. 1=Stutter mode enabled. |
| 15:12 | Reserved. |
| 11 | **RengExecuteOnRegUpdate**. Read-write. Reset: 0. 1=Enable saving GMC registers prior to power-down from power gating. |
| 10 | Reserved. |
| 9:0 | **RengExecuteNonsecureStartPtr**. IF (CriticalRegsLock == 0) THEN Read-write. ELSE Read-only. ENDIF. Reset: 0. This pointer splits the register engine RAM into secure and non-secure regions. The RAM cannot be written below this pointer. |

### GMMx2C04 HDP Default Surface Base Address

| Bits | Description |
|---|---|
| 31:28 | Reserved. |
| 27:0 | **NonsurfBase**. Read-write. Reset: 0. Specifies GPU address bits 35:8 of the default surface base address. |

### GMMx5428 Configuration Memory Size

Scratch register that BIOS uses to pass the memory configuration to the GPU driver.

| Bits | Description |
|---|---|
| 31:0 | **ConfigMemsize**. Read-write. Reset: 0. Configuration memory size. |

### GMMx5490 Frame Buffer Access Control

| Bits | Description |
|---|---|
| 31:2 | Reserved. |
| 1 | **FbWriteEn**. Read-write. Reset: 0. Enables host writes to the Frame Buffer. |
| 0 | **FbReadEn**. Read-write. Reset: 0. Enables host reads to the Frame Buffer. |

### GMMx6[3FC,12C:110] Display Output Scratch

| Bits | Description |
|---|---|
| 31:0 | **DoutScratch**. Read-write. Reset: 0. |

### GMMx[79,6D]F0 Display Controller Front-end (DCFE) Clock Control

| Bits | Description |
|---|---|
| 31:13 | Reserved. |
| 12 | **CrtcDispclkGSclGateDisable**. Read-write. Reset: 0. BIOS: 0. Disable clock gating for dispclk_g_scl. See 2.5.5.3 [GPU and Root Complex Clock Gating]. |
| 11:9 | Reserved. |
| 8 | **CrtcDispclkGDcpGateDisable**. Read-write. Reset: 0. BIOS: 0. Disable clock gating for dispclk_g_dcp. See 2.5.5.3 [GPU and Root Complex Clock Gating]. |
| 7:5 | Reserved. |
| 4 | **CrtcDispclkRDcfeGateDisable**. Read-write. Reset: 0. BIOS: 0. Disable clock gating for dispclk_r. See 2.5.5.3 [GPU and Root Complex Clock Gating]. |
| 3:0 | Reserved. |

## 3.18 APIC Registers

See 3.1 [Register Descriptions and Mnemonics] for a description of the register naming convention.

### APIC20 APIC ID Register

| Bits | Description |
|---|---|
| 31:24 | **ApicId**. Read-write. Reset: {000000b, CpuCoreNum}. BIOS: See 2.4.6.1.1. When D18F0x68[ApicExtId and ApicExtBrdCst] = 11b, all 8 bits of this field are used; if either of these bits is low, then bits[3:0] of this field are used and bits[7:4] are reserved. See 2.4.2 [Processor Cores and Downcoring]. |
| 23:0 | Reserved. |

### APIC30 APIC Version Register

Reset: 80xx_0010h.

| Bits | Description |
|---|---|
| 31 | **ExtApicSpace: extended APIC register space present**. Read-only. This bit indicates the presence of extended APIC register space starting at APIC400. |
| 30:24 | Reserved. |
| 23:16 | **MaxLvtEntry**. Read-only. Reset: Product-specific.. This field specifies the number of entries in the local vector table minus one. |
| 15:8 | Reserved. |
| 7:0 | **Version**. Read-only. This field indicates the version number of this APIC implementation. |

### APIC80 Task Priority Register

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:8 | Reserved. |
| 7:0 | **Priority**. Read-write. This field is assigned by software to set a threshold priority at which the core is interrupted. |

### APIC90 Arbitration Priority Register

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:8 | Reserved. |
| 7:0 | **Priority**. Read-only. This field indicates the current priority for a pending interrupt, or a task or interrupt being serviced by the core. The priority is used to arbitrate between cores to determine which accepts a lowest-priority interrupt request. |

### APICA0 Processor Priority Register

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:8 | Reserved. |
| 7:0 | **Priority**. Read-only. This field indicates the core's current priority servicing a task or interrupt, and is used to determine if any pending interrupts should be serviced. It is the higher value of the task priority value and the current highest in-service interrupt. |

### APICB0 End of Interrupt Register

This register is written by the software interrupt handler to indicate the servicing of the current interrupt is complete.

| Bits | Description |
|---|---|
| 31:0 | Reserved. Write-only. Reads return undefined data. |

### APICC0 Remote Read Register

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:0 | **RemoteReadData**. Read-only. This field contains the data resulting from a valid completion of a remote read inter-processor interrupt. |

### APICD0 Logical Destination Register

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:24 | **Destination**. Read-write. This field contains this APIC's destination identification. This field is used to determine which interrupts should be accepted. |
| 23:0 | Reserved. |

### APICE0 Destination Format Register

Reset: FFFF_FFFFh.

| Bits | Description |
|---|---|
| 31:28 | **Format**. Read-write. This field controls which format to use when accepting interrupts with a logical destination mode. The allowed values are:<br>Bits — Definition<br>0h — Cluster destinations are used<br>Fh — Flat destinations are used |
| 27:0 | Reserved. |

### APICF0 Spurious Interrupt Vector Register

Reset: 0000_00FFh.

| Bits | Description |
|---|---|
| 31:10 | Reserved. |
| 9 | **FocusDisable**. Read-write. 1=Disable focus core checking during lowest-priority arbitrated interrupts. |
| 8 | **APICSWEn: APIC software enable**. Read-write. 0=SMI, NMI, INIT, Startup, and Remote Read interrupts may be accepted; pending interrupts in the APIC[170:100] and APIC[270:200] are held, but further fixed, lowest-priority, LINT, and ExtInt interrupts are not accepted. All LVT entry mask bits are set and cannot be cleared. |
| 7:0 | **Vector**. Read-write. This field contains the vector that is sent to the core in the event of a spurious interrupt. The behavior of bits 3:0 are controlled as specified by [The Link Transaction Control Register] D18F0x68[ApicExtSpur]. |

### APIC[170:100] In-Service Registers

Reset: 0000_0000h.

The in-service registers provide a bit per interrupt to indicate that the corresponding interrupt is being serviced by the core. APIC100[15:0] are reserved.

| Register | Function | Register | Function |
|---|---|---|---|
| APIC100 | Interrupts 31-16 | APIC140 | Interrupts 159-128 |
| APIC110 | Interrupts 63-32 | APIC150 | Interrupts 191-160 |
| APIC120 | Interrupts 95-64 | APIC160 | Interrupts 223-192 |
| APIC130 | Interrupts 127-96 | APIC170 | Interrupts 255-224 |

| Bits | Description |
|---|---|
| 31:0 | **InServiceBits**. Read-only. These bits are set when the corresponding interrupt is being serviced by the core. |

### APIC[1F0:180] Trigger Mode Registers

Reset: 0000_0000h.

The trigger mode registers provide a bit per interrupt to indicate that the assertion mode of each interrupt. APIC180[15:0] are reserved.

| Register | Function | Register | Function |
|---|---|---|---|
| APIC180 | Interrupts 31-16 | APIC1C0 | Interrupts 159-128 |
| APIC190 | Interrupts 63-32 | APIC1D0 | Interrupts 191-160 |
| APIC1A0 | Interrupts 95-64 | APIC1E0 | Interrupts 223-192 |
| APIC1B0 | Interrupts 127-96 | APIC1F0 | Interrupts 255-224 |

| Bits | Description |
|---|---|
| 31:0 | **TriggerModeBits**. Read-only. The corresponding trigger mode bit is updated when an interrupt enters servicing. The values are:<br>• 0b = edge-triggered interrupt.<br>• 1b = level-triggered interrupt. |

### APIC[270:200] Interrupt Request Registers

Reset: 0000_0000h.

The interrupt request registers provide a bit per interrupt to indicate that the corresponding interrupt has been accepted by the APIC. APIC200[15:0] are reserved.

| Register | Function | Register | Function |
|---|---|---|---|
| APIC200 | Interrupts 31-16 | APIC240 | Interrupts 159-128 |
| APIC210 | Interrupts 63-32 | APIC250 | Interrupts 191-160 |
| APIC220 | Interrupts 95-64 | APIC260 | Interrupts 223-192 |
| APIC230 | Interrupts 127-96 | APIC270 | Interrupts 255-224 |

| Bits | Description |
|---|---|
| 31:0 | **RequestBits**. Read-only. The corresponding request bit is set when an interrupt is accepted by the APIC. |

### APIC280 Error Status Register

Reset: 0000_0000h.

Writes to this register trigger an update of the register state. The value written by software is arbitrary. Each write causes the internal error state to be loaded into this register, clearing the internal error state. Consequently, a second write prior to the occurrence of another error causes the register to be overwritten with cleared data.

| Bits | Description |
|---|---|
| 31:8 | Reserved. |
| 7 | **IllegalRegAddr: illegal register address**. Read-write. This bit indicates that an access to a nonexistent register location within this APIC was attempted. |
| 6 | **RcvdIllegalVector: received illegal vector**. Read-write. This bit indicates that this APIC has received a message with an illegal vector (00h to 0Fh for fixed and lowest priority interrupts). |
| 5 | **SentIllegalVector**. Read-write. This bit indicates that this APIC attempted to send a message with an illegal vector (00h to 0Fh for fixed and lowest priority interrupts). |
| 4 | Reserved. |
| 3 | **RcvAcceptError: receive accept error**. Read-write. This bit indicates that a message received by this APIC was not accepted by this or any other APIC. |
| 2 | **SendAcceptError**. Read-write. This bit indicates that a message sent by this APIC was not accepted by any APIC. |
| 1:0 | Reserved. |

**APIC300 Interrupt Command Register Low**

Reset: 0000_0000h.

Not all combinations of ICR fields are valid. Only the following combinations are valid:

| Message Type | Trigger Mode | Level | Destination Shorthand |
|---|---|---|---|
| Fixed | Edge | N/A | N/A |
| | Level | Assert | N/A |
| Lowest Priority, SMI, NMI, INIT | Edge | N/A | Destination or all excluding self. |
| | Level | Assert | Destination or all excluding self |
| Startup | N/A | N/A | Destination or all excluding self |

| Bits | Description |
|---|---|
| 31:20 | Reserved. |
| 19:18 | **DestShrthnd: destination shorthand**. Read-write. This field provides a quick way to specify a destination for a message.<br>Bits — Definition<br>00b — Destination field<br>01b — Self<br>10b — All including self<br>11b — All excluding self (This sends a message with a destination encoding of all 1s, so if lowest priority is used the message could end up being reflected back to this APIC.)<br>If all including self or all excluding self is used, then destination mode is ignored and physical is automatically used. |
| 17:16 | **RemoteRdStat: remote read status**. Read-only.<br>Bits — Definition<br>00b — Read was invalid<br>01b — Delivery pending<br>10b — Delivery done and access was valid<br>11b — Reserved |
| 15 | **TM: trigger mode**. Read-write. This bit indicates how this interrupt is triggered. 1 = Level triggered. 0 = Edge triggered. |
| 14 | **Level**. Read-write. 0=Deasserted. 1 = Asserted. |
| 13 | Reserved. |
| 12 | **DlvryStat: delivery status**. Read-only. This bit is set to indicate that the interrupt has not yet been accepted by the destination core(s). |
| 11 | **DM: destination mode**. Read-write. 0 = Physical. 1 = Logical. |

<table>
<tr><td>10:8</td><td>MsgType: message type. Read-write.
<table>
<tr><th>Bits</th><th>Definition</th><th>Bits</th><th>Definition</th></tr>
<tr><td>000b</td><td>Fixed</td><td>100b</td><td>NMI</td></tr>
<tr><td>001b</td><td>Lowest Priority</td><td>101b</td><td>INIT</td></tr>
<tr><td>010b</td><td>SMI</td><td>110b</td><td>Startup</td></tr>
<tr><td>011b</td><td>Remote read</td><td>111b</td><td>External interrupt</td></tr>
</table></td></tr>
<tr><td>7:0</td><td>Vector. Read-write. This field contains the vector that is sent for this interrupt source.</td></tr>
</table>

### APIC310 Interrupt Command Register High

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:24 | **DestinationField**. Read-write. This field contains the destination encoding used when APIC300[DestShrthnd] is 00b. |
| 23:0 | Reserved. |

### APIC320 Timer Local Vector Table Entry

Reset: 0001_0000h.

| Bits | Description |
|---|---|
| 31:18 | Reserved. |
| 17 | **Mode**. Read-write. 1 = Periodic. 0 = One-shot |
| 16 | **Mask**. Read-write. If this bit is set, this local vector table entry does not generate interrupts. |
| 15:13 | Reserved. |
| 12 | **DlvryStat: delivery status**. Read-only. This bit is set to indicate that the interrupt has not yet been accepted by the core. |
| 11 | Reserved. |
| 10:8 | **MsgType: message type**. Write-only. Read always returns 000b. See 2.4.6.1.12 [Generalized Local Vector Table] for supported message types. |
| 7:0 | **Vector**. Read-write. This field contains the vector that is sent for this interrupt source. |

### APIC330 Thermal Local Vector Table Entry

Reset: 0001_0000h. Interrupts for this local vector table are caused by changes to the current P-state limit. This includes changes due to the following:

- 2.10.3.1 [PROCHOT_L and Hardware Thermal Control (HTC)].

| Bits | Description |
|---|---|
| 31:17 | Reserved. |
| 16 | **Mask**. Read-write. If this bit is set, this local vector table entry does not generate interrupts. |
| 15:13 | Reserved. |
| 12 | **DlvryStat: delivery status**. Read-only. This bit is set to indicate that the interrupt has not yet been accepted by the core. |

| | |
|---|---|
| 11 | Reserved. |
| 10:8 | **MsgType: message type**. Read-write. See 2.4.6.1.12 [Generalized Local Vector Table] for supported message types. |
| 7:0 | **Vector**. Read-write. This field contains the vector that is sent for this interrupt source. |

**APIC340 Performance Counter Vector Table Entry**

Reset: 0001_0000h.

Interrupts for this local vector table are caused by overflows of [The Performance Event Counter Registers (PERF_CTR[3:0])] MSRC001_00[07:04].

| Bits | Description |
|---|---|
| 31:17 | Reserved. |
| 16 | **Mask**. Read-write. If this bit is set, this local vector table entry does not generate interrupts. |
| 15:13 | Reserved. |
| 12 | **DlvryStat: delivery status**. Read-only. This bit is set to indicate that the interrupt has not yet been accepted by the core. |
| 11 | Reserved. |
| 10:8 | **MsgType: message type**. Read-write. See 2.4.6.1.12 [Generalized Local Vector Table] for supported message types. |
| 7:0 | **Vector**. Read-write. This field contains the vector that is sent for this interrupt source. |

**APIC350 Local Interrupt 0 (Legacy INTR) Local Vector Table Entry**

Reset: 0001_0000h.

| Bits | Description |
|---|---|
| 31:17 | Reserved. |
| 16 | **Mask**. Read-write. If this bit is set, this local vector table entry does not generate interrupts. |
| 15 | **TM: trigger mode**. Read-write. This bit indicates how this interrupt is triggered. It is defined as follows:<br>• 0 = Edge triggered<br>• 1 = Level triggered |
| 14 | **RmtIRR**. Read-only. If trigger mode is level, remote IRR is set when the interrupt has begun service. Remote IRR is cleared when the end of interrupt has occurred. |
| 13 | **PinPol: pin polarity**. Read-write. This bit is not used because LINT interrupts are delivered by link messages instead of individual pins. |
| 12 | **DlvryStat: delivery status**. Read-only. This bit is set to indicate that the interrupt has not yet been accepted by the core. |
| 11 | Reserved. |
| 10:8 | **MsgType: message type**. Read-write. See 2.4.6.1.12 [Generalized Local Vector Table] for supported message types. |
| 7:0 | **Vector**. Read-write. This field contains the vector that is sent for this interrupt source. |

### APIC360 Local Interrupt 1(Legacy NMI) Local Vector Table Entry

Reset: 0001_0000h.

| Bits | Description |
|---|---|
| 31:0 | See: APIC350. |

### APIC370 Error Local Vector Table Entry

Reset: 0001_0000h.

| Bits | Description |
|---|---|
| 31:17 | Reserved. |
| 16 | **Mask**. Read-write. If this bit is set, this local vector table entry does not generate interrupts. |
| 15:13 | Reserved. |
| 12 | **DlvryStat: delivery status**. Read-only. This bit is set to indicate that the interrupt has not yet been accepted by the core. |
| 11 | Reserved. |
| 10:8 | **MsgType: message type**. RAZ; write. See 2.4.6.1.12 [Generalized Local Vector Table] for supported message types. |
| 7:0 | **Vector**. Read-write. This field contains the vector that is sent for this interrupt source. |

### APIC380 Timer Initial Count Register

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:0 | **Count**. Read-write. This field contains the value copied into the current count register when the timer is loaded or reloaded. |

### APIC390 Timer Current Count Register

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:0 | **Count**. Read-only. This field contains the current value of the counter. |

### APIC3E0 Timer Divide Configuration

The Div bits are encoded as follows:

**Table 139: Div[3,1:0] Value Table**

| Div[3] | Div[1:0] | Resulting Timer Divide |
|---|---|---|
| 0 | 00b | 2 |
| 0 | 01b | 4 |
| 0 | 10b | 8 |
| 0 | 11b | 16 |

**Table 139: Div[3,1:0] Value Table**

| Div[3] | Div[1:0] | Resulting Timer Divide |
|---|---|---|
| 1 | 00b | 32 |
| 1 | 01b | 64 |
| 1 | 10b | 128 |
| 1 | 11b | 1 |

| Bits | Description |
|---|---|
| 31:4 | RAZ. |
| 3 | **Div[3]**. Read-write. Reset: 0. See Table 139. |
| 2 | RAZ. |
| 1:0 | **Div[1:0]**. Read-write. Reset: 0. See Table 139. |

### APIC400 Extended APIC Feature Register

| Bits | Description |
|---|---|
| 31:24 | Reserved. |
| 23:16 | **ExtLvtCount: extended local vector table count**. Read-only. Reset: 04h. This specifies the number of extended LVT registers in the local APIC. These registers are [The Extended Interrupt [3:0] Local Vector Table Registers] APIC[530:500]. |
| 15:3 | Reserved. |
| 2 | **ExtApicIdCap: extended APIC ID capable**. Read-only. Reset: 1. Indicates that the processor is capable of supporting an 8-bit APIC ID, controlled by APIC410[ExtApicIdEn]. |
| 1 | **SeioCap: specific end of interrupt capable**. Read-only. Reset: 1. This bit indicates that the [The Specific End Of Interrupt Register] APIC420 is present. |
| 0 | **IerCap: interrupt enable register capable**. Read-only. Reset: 1. This bit indicates that the [The Interrupt Enable] APIC[4F0:480] are present. See 2.4.6.1.6 [Interrupt Masking]. |

### APIC410 Extended APIC Control Register

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:3 | Reserved. |
| 2 | **ExtApicIdEn: extended APIC ID enable**. Read-write. 1=Enable 8-bit APIC ID; APIC20[ApicId] supports an 8-bit value; an interrupt broadcast in physical destination mode requires that the IntDest[7:0]=1111_1111 (instead of xxxx_1111); a match in physical destination mode occurs when (IntDest[7:0] == ApicId[7:0]) instead of (IntDest[3:0] == ApicId[3:0]). Extended APIC ID can also be enabled by writing D18F0x68[ApicExtId] and D18F0x68[ApicExtBrdCst]. If this bit is set, then D18F0x68[ApicExtId] and D18F0x68[ApicExtBrdCst] must be set. |
| 1 | **SeoiEn**. Read-write. This bit enables SEOI generation when a write to the specific end of interrupt register is received. |
| 0 | **IerEn**. Read-write. This bit enables writes to the interrupt enable registers. |

### APIC420 Specific End Of Interrupt Register

Reset: 0000_0000h.

| Bits | Description |
|---|---|
| 31:8 | Reserved. |
| 7:0 | **EoiVec: end of interrupt vector**. Read-write. A write to this field causes an end of interrupt cycle to be performed for the vector specified in this field. The behavior is undefined if no interrupt is pending for the specified interrupt vector. |

### APIC[4F0:480] Interrupt Enable

Reset: FFFF_FFFFh.

| Register | Function | Register | Function |
|---|---|---|---|
| APIC480 | Interrupts 31-0 | APIC4C0 | Interrupts 159-128 |
| APIC490 | Interrupts 63-32 | APIC4D0 | Interrupts 191-160 |
| APIC4A0 | Interrupts 95-64 | APIC4E0 | Interrupts 223-192 |
| APIC4B0 | Interrupts 127-96 | APIC4F0 | Interrupts 255-224 |

| Bits | Description |
|---|---|
| 31:0 | **InterruptEnableBits**. Read-write. The interrupt enable bits can be used to enable each of the 256 interrupts. |

### APIC[530:500] Extended Interrupt [3:0] Local Vector Table Registers

Reset: 0001_0000h. These registers provide additional local vector table entries for selected internal interrupt sources.

| Bits | Description |
|---|---|
| 31:17 | Reserved. |
| 16 | **Mask**. Read-write. 1=This LVT entry does not generate interrupts. |
| 15:13 | Reserved. |
| 12 | **DlvryStat: delivery status**. Read-only. 1=The interrupt has not yet been accepted by the CPU. |
| 11 | Reserved. |
| 10:8 | **MsgType: message type**. Read-write. Specifies the interrupt type generated by this LVT entry. See 2.4.6.1.12 [Generalized Local Vector Table] for supported message types. |
| 7:0 | **Vector**. Read-write. This field contains the vector generated by this LVT entry. |

## 3.19 CPUID Instruction Registers

Processor feature capabilities and configuration information are provided through the CPUID instruction. Different information is accessed by (1) setting EAX as an index to the registers to be read, (2) executing the CPUID instruction, and (3) reading the results in EAX, EBX, ECX, and EDX. The phrase *CPUID function X* or *CPUID FnX* refers to the CPUID instruction when EAX is preloaded with X. Undefined function numbers return 0's in all 4 registers. See 2.4.8 [CPUID Instruction].

Unless otherwise specified, single-bit feature fields are encoded as 1=Feature is supported by the processor; 0=Feature is not supported by the processor.

The following provides AMD family 12h processor specific details about CPUID. See the *CPUID Specification* for further information.

### CPUID Fn0000_0000_EAX Processor Vendor and Largest Standard Function Number

| Bits | Description |
|---|---|
| 31:0 | **LFuncStd: largest standard function**. Value: 0000_0006h. The largest CPUID standard function input value supported by the processor implementation. |

### CPUID Fn0000_0000_E[B,C,D]X Processor Vendor and Largest Standard Function Number

**Table 140: Reset mapping for** CPUID Fn0000_0000_E[B,C,D]X**.**

| Register | Value | Description |
|---|---|---|
| CPUID Fn0000_0000_EBX | 6874_7541h | The ASCII characters: h, t, u, A |
| CPUID Fn0000_0000_ECX | 444D_4163h | The ASCII characters: D, M, A, c |
| CPUID Fn0000_0000_EDX | 6974_6E65h | The ASCII characters: i, t, n, e |

| Bits | Description |
|---|---|
| 31:0 | **Vendor: vendor**. The 12 8-bit ASCII character codes to create the string “AuthenticAMD”. |

### CPUID Fn0000_0001_EAX Family, Model, Stepping Identifiers

This register provides identical information to D18F3xFC.

**Family** is an 8-bit value and is defined as: **Family[7:0] =** ({0000b,BaseFamily[3:0]} + ExtendedFamily[7:0]). E.g. If BaseFamily[3:0]=Fh and ExtendedFamily[7:0]=03h, then Family[7:0]=12h. This document applies only to family 12h processors.

**Model** is an 8-bit value and is defined as: **Model[7:0]** = {ExtendedModel[3:0], BaseModel[3:0]}. E.g. If ExtendedModel[3:0]=Eh and BaseModel[3:0]=8h, then Model[7:0] = E8h. Model numbers vary with product.

| Bits | Description |
|---|---|
| 31:28 | Reserved. |
| 27:20 | **ExtendedFamily: extended family**. Value: 3h. |
| 19:16 | **ExtendedModel: extended model**. Value: Product-specific. |
| 15:12 | Reserved. |
| 11:8 | **BaseFamily: base family**. Value: Fh. |
| 7:4 | **BaseModel: base model**. Value: Product-specific. |
| 3:0 | **Stepping: processor stepping (revision) for a specific model**. Value: Product-specific. |

**CPUID Fn0000_0001_EBX LocalApicId, LogicalProcessorCount, CLFlush, 8BitBrandId**

| Bits | Description |
|---|---|
| 31:24 | **LocalApicId: initial local APIC physical ID**. Provides the initial APIC20[ApicId] value. Changes to APIC20[ApicId] do not affect the value of this CPUID register. See 2.4.2 [Processor Cores and Downcoring]. |
| 23:16 | **LogicalProcessorCount: logical processor count**. Value: Product-specific. IF (CPUID Fn0000_0001_EDX[HTT] == 1) THEN this field indicates the number of cores in the processor as CPUID Fn8000_0008_ECX[NC] + 1 ELSE Reserved. ENDIF. |
| 15:8 | **CLFlush: CLFLUSH size in quadwords**. Value: 08h. |
| 7:0 | **8BitBrandId: 8 bit brand ID**. Value: 00h. Indicates that the brand ID is in CPUID Fn8000_0001_EBX. |

**CPUID Fn0000_0001_ECX Feature Identifiers**

| Bits | Description |
|---|---|
| 31 | RAZ. Reserved for use by hypervisor to indicate guest status. |
| 30:24 | Reserved. |
| 23 | **POPCNT: POPCNT instruction**. Value: 1. |
| 22:14 | Reserved. |
| 13 | **CMPXCHG16B: CMPXCHG16B instruction**. Value: 1. |
| 12:4 | Reserved. |
| 3 | **Monitor: Monitor/Mwait instructions**. IF (MSRC001_0015[MonMwaitDis]==0)THEN Value: 1 ELSE Value: 0 ENDIF. |
| 2:1 | Reserved. |
| 0 | **SSE3: SSE3 extensions**. IF (MSRC001_0015[SseDis]==0)THEN Value: 1 ELSE Value: 0 ENDIF. |

**CPUID Fn0000_0001_EDX Feature Identifiers**

| Bits | Description |
|---|---|
| 31:29 | Reserved. |
| 28 | **HTT: hyper-threading technology**. Value: Product-specific. This bit qualifies the meaning of CPUID Fn0000_0001_EBX[LogicalProcessorCount]. 1=Multi core product (CPUID Fn8000_0008_ECX[NC] != 0). 0=Single core product (CPUID Fn8000_0008_ECX[NC] = 0). |
| 27 | Reserved. |
| 26 | **SSE2: SSE2 extensions**. IF (MSRC001_0015[SseDis]==0) THEN Value: 1 ELSE Value: 0 ENDIF. |
| 25 | **SSE: SSE extensions**. IF (MSRC001_0015[SseDis]==0) THEN Value: 1 ELSE Value: 0 ENDIF. |
| 24 | **FXSR: FXSAVE and FXRSTOR instructions**. Value: 1. |
| 23 | **MMX: MMX™ instructions**. Value: 1. |
| 22:20 | Reserved. |

| Bits | Description |
|---|---|
| 19 | **CLFSH: CLFLUSH instruction**. Value: 1. |
| 18 | Reserved. |
| 17 | **PSE36: page-size extensions**. Value: 1. |
| 16 | **PAT: page attribute table**. Value: 1. |
| 15 | **CMOV: conditional move instructions, CMOV, FCOMI, FCMOV**. Value: 1. |
| 14 | **MCA: machine check architecture, MCG_CAP**. Value: 1. |
| 13 | **PGE: page global extension, CR4.PGE**. Value: 1. |
| 12 | **MTRR: memory-type range registers**. Value: 1. |
| 11 | **SysEnterSysExit: SYSENTER and SYSEXIT instructions**. Value: 1. |
| 10 | Reserved. |
| 9 | **APIC: advanced programmable interrupt controller (APIC) exists and is enabled**. Value: MSR0000_001B[ApicEn]. |
| 8 | **CMPXCHG8B: CMPXCHG8B instruction**. Value: 1. |
| 7 | **MCE: machine check exception, CR4.MCE**. Value: 1. |
| 6 | **PAE: physical-address extensions (PAE)**. Value: 1. |
| 5 | **MSR: AMD model-specific registers (MSRs), with RDMSR and WRMSR instructions**. Value: 1. |
| 4 | **TSC: time stamp counter, RDTSC/RDTSCP instructions, CR4.TSD**. Value: 1. |
| 3 | **PSE: page-size extensions (4 MB pages)**. Value: 1. |
| 2 | **DE: debugging extensions, IO breakpoints, CR4.DE**. Value: 1. |
| 1 | **VME: virtual-mode enhancements**. Value: 1. |
| 0 | **FPU: x87 floating point unit on-chip**. Value: 1. |

### CPUID Fn0000_000[4,3,2] Reserved

| Bits | Description |
|---|---|
| 31:0 | Reserved. |

### CPUID Fn0000_0005_EAX Monitor/MWait

| Bits | Description |
|---|---|
| 31:16 | Reserved. |
| 15:0 | **SMon: Smallest monitor-line size in bytes**. Value: 40h. |

### CPUID Fn0000_0005_EBX Monitor/MWait

| Bits | Description |
|---|---|
| 31:16 | Reserved. |
| 15:0 | **LMon: Largest monitor-line size in bytes**. Value: 40h. |

**CPUID Fn0000_0005_ECX Monitor/MWait**

| Bits | Description |
|---|---|
| 31:2 | Reserved. |
| 1 | **IBE: Interrupt break-event**. Value: 1. |
| 0 | **EMX: Enumerate MONITOR/MWAIT extensions**. Value: 1. |

**CPUID Fn0000_0005_EDX Monitor/MWait**

| Bits | Description |
|---|---|
| 31:0 | Reserved. |

**CPUID Fn0000_0006_EAX Power Management Features**

| Bits | Description |
|---|---|
| 31:0 | Reserved. |

**CPUID Fn0000_0006_EBX Power Management Features**

| Bits | Description |
|---|---|
| 31:0 | Reserved. |

**CPUID Fn0000_0006_ECX Power Management Features**

| Bits | Description |
|---|---|
| 31:1 | Reserved. |
| 0 | **EffFreq: effective frequency interface**. Value: 1. |

**CPUID Fn0000_0006_EDX Power Management Features**

| Bits | Description |
|---|---|
| 31:0 | Reserved. |

**CPUID Fn8000_0000_EAX Processor Vendor and Largest Extended Function Number**

| Bits | Description |
|---|---|
| 31:0 | **LFuncExt: largest extended function**. Value: 8000_001Bh. The largest CPUID extended function input value supported by the processor implementation. |

### CPUID Fn8000_0000_E[B,C,D]X Processor Vendor and Largest Extended Function Number

**Table 141: Reset mapping for** CPUID Fn8000_0000_EAX**.**

| Register | Value | Description |
|---|---|---|
| CPUID Fn8000_0000_EBX | 6874_7541h | The ASCII characters: h, t, u, A |
| CPUID Fn8000_0000_ECX | 444D_4163h | The ASCII characters: D, M, A, c |
| CPUID Fn8000_0000_EDX | 6974_6E65h | The ASCII characters: i, t, n, e |

| Bits | Description |
|---|---|
| 31:0 | **Vendor: vendor**. The 12 8-bit ASCII character codes to create the string “AuthenticAMD”. |

### CPUID Fn8000_0001_EAX Family, Model, Stepping Identifiers

See CPUID Fn0000_0001_EAX.

| Bits | Description |
|---|---|
| 31:28 | Reserved. |
| 27:20 | **ExtendedFamily: extended family**. Value: 03h. |
| 19:16 | **ExtendedModel: extended model**. Value: Product-specific. |
| 15:12 | Reserved. |
| 11:8 | **BaseFamily: base family**. Value: 0Fh. |
| 7:4 | **BaseModel: base model**. Value: Product-specific. |
| 3:0 | **Stepping: processor stepping (revision) for a specific model**. Value: Product-specific. |

### CPUID Fn8000_0001_EBX BrandId Identifier

| Bits | Description |
|---|---|
| 31:28 | **PkgType: package type**. Value: Product-specific. Specifies the package type. This field is encoded as follows:<br>Bits — Description<br>0000b — Reserved<br>0001b — FS1 (uPGA)<br>0010b — FM1 (PGA) |
| 27:16 | Reserved. |
| 15:0 | **BrandId: brand ID**. Value: D18F3x1F0[BrandId]. |

### CPUID Fn8000_0001_ECX Feature Identifiers

| Bits | Description |
|---|---|
| 31:14 | Reserved. |
| 13 | **WDT: watchdog timer support**. Value: 1. |
| 12 | **SKINIT: SKINIT and STGI support**. Value: 1. |

| Bits | Description |
|---|---|
| 11 | Reserved. |
| 10 | **IBS: Instruction Based Sampling**. Value: 1. |
| 9 | **OSVW: OS Visible Work-around support**. Value: 1. |
| 8 | **3DNowPrefetch: Prefetch and PrefetchW instructions**. Value: 1. |
| 7 | **MisAlignSse: Misaligned SSE Mode**. IF (MSRC001_0015[MisAlignSseDis]==0) THEN Value: 1 ELSE Value: 0 ENDIF. |
| 6 | **SSE4A: EXTRQ, INSERTQ, MOVNTSS, and MOVNTSD instruction support**. IF (MSRC001_0015[SseDis]==0) THEN Value: 1 ELSE Value: 0 ENDIF. |
| 5 | **ABM: advanced bit manipulation**. Value: Product-specific. LZCNT instruction support. |
| 4 | **AltMovCr8: LOCK MOV CR0 means MOV CR8**. Value: 1. |
| 3 | **ExtApicSpace: extended APIC register space**. Value: 1. |
| 2 | **SVM: Secure Virtual Mode feature**. Value: Product-specific. Indicates support for: VMRUN, VMLOAD, VMSAVE, CLGI, VMMCALL, and INVLPGA. |
| 1 | **CmpLegacy: core multi-processing legacy mode**. Value: Product-specific. 1=Multi core product (CPUID Fn8000_0008_ECX[NC] != 0). 0=Single core product (CPUID Fn8000_0008_ECX[NC] = 0). |
| 0 | **LahfSahf: LAHF/SAHF instructions**. Value: 1. |

**CPUID Fn8000_0001_EDX Feature Identifiers**

| Bits | Description |
|---|---|
| 31 | **3DNow: 3DNow!™ instructions**. Value: 1. |
| 30 | **3DNowExt: AMD extensions to 3DNow!™ instructions**. Value: 1. |
| 29 | **LM: long mode**. May vary by product. Value: Product-specific. |
| 28 | Reserved. |
| 27 | **RDTSCP: RDTSCP instruction**. Value: 1. |
| 26 | **Page1GB: one GB large page support**. Value: 1. |
| 25 | **FFXSR: FXSAVE and FXRSTOR instruction optimizations**. Value: 1. |
| 24 | **FXSR: FXSAVE and FXRSTOR instructions**. Value: 1. |
| 23 | **MMX: MMX™ instructions**. Value: 1. |
| 22 | **MmxExt: AMD extensions to MMX™ instructions.** Value: 1. |
| 21 | Reserved. |
| 20 | **NX: no-execute page protection**. Value: 1. |
| 19:18 | Reserved. |
| 17 | **PSE36: page-size extensions**. Value: 1. |
| 16 | **PAT: page attribute table**. Value: 1. |
| 15 | **CMOV: conditional move instructions, CMOV, FCOMI, FCMOV**. Value: 1. |
| 14 | **MCA: machine check architecture, MCG_CAP**. Value: 1. |
| 13 | **PGE: page global extension, CR4.PGE**. Value: 1. |

| Bits | Description |
|---|---|
| 12 | **MTRR: memory-type range registers**. Value: 1. |
| 11 | **SysCallSysRet: SYSCALL and SYSRET instructions**. Value: 1. |
| 10 | Reserved. |
| 9 | **APIC: advanced programmable interrupt controller (APIC) exists and is enabled**. Value: MSR0000_001B[ApicEn]. |
| 8 | **CMPXCHG8B: CMPXCHG8B instruction**. Value: 1. |
| 7 | **MCE: machine check exception, CR4.MCE**. Value: 1. |
| 6 | **PAE: physical-address extensions (PAE)**. Value: 1. |
| 5 | **MSR: AMD model-specific registers (MSRs), with RDMSR and WRMSR instructions**. Value: 1. |
| 4 | **TSC: time stamp counter, RDTSC/RDTSCP instructions, CR4.TSD**. Value:1. |
| 3 | **PSE: page-size extensions (4 MB pages)**. Value: 1. |
| 2 | **DE: debugging extensions, IO breakpoints, CR4.DE**. Value: 1. |
| 1 | **VME: virtual-mode enhancements**. Value: 1. |
| 0 | **FPU: x87 floating point unit on-chip**. Value: 1. |

### CPUID Fn8000_000[4,3,2]_E[D,C,B,A]X Processor Name String Identifier

**Table 142: Reset mapping for** CPUID Fn8000_000[4,3,2]_E[D,C,B,A]X**.**

| Register | Value | Register | Value |
|---|---|---|---|
| CPUID Fn8000_0002_EAX | MSRC001_0030[31:0] | CPUID Fn8000_0003_ECX | MSRC001_0033[31:0] |
| CPUID Fn8000_0002_EBX | MSRC001_0030[63:32] | CPUID Fn8000_0003_EDX | MSRC001_0033[63:32] |
| CPUID Fn8000_0002_ECX | MSRC001_0031[31:0] | CPUID Fn8000_0004_EAX | MSRC001_0034[31:0] |
| CPUID Fn8000_0002_EDX | MSRC001_0031[63:32] | CPUID Fn8000_0004_EBX | MSRC001_0034[63:32] |
| CPUID Fn8000_0003_EAX | MSRC001_0032[31:0] | CPUID Fn8000_0004_ECX | MSRC001_0035[31:0] |
| CPUID Fn8000_0003_EBX | MSRC001_0032[63:32] | CPUID Fn8000_0004_EDX | MSRC001_0035[63:32] |

| Bits | Description |
|---|---|
| 31:0 | **ProcName: processor name**. These return the ASCII string corresponding to the processor name, stored in [The Processor Name String Registers] MSRC001_00[35:30]. |

### CPUID Fn8000_0005_EAX L1 TLB Identifiers

This provides the processor's first level cache and TLB characteristics for each core. The *associativity* fields returned are encoded as follows:

00h Reserved.
01h Direct mapped.
02h - FEh Specifies the associativity; e.g., 04h would indicate a 4-way associativity.
FFh - Fully associative.

| Bits | Description |
|---|---|
| 31:24 | **L1DTlb2and4MAssoc: data TLB associativity for 2 MB and 4 MB pages**. Value: 0FFh. |

| | |
|---|---|
| 23:16 | **L1DTlb2and4MSize: data TLB number of entries for 2 MB and 4 MB pages**. Value: 48. The value returned is for the number of entries available for the 2 MB page size; 4 MB pages require two 2 MB entries, so the number of entries available for the 4 MB page size is one-half the returned value. |
| 15:8 | **L1ITlb2and4MAssoc: instruction TLB associativity for 2 MB and 4 MB pages**. Value: 0FFh. |
| 7:0 | **L1ITlb2and4MSize: instruction TLB number of entries for 2 MB and 4 MB pages**. Value: 16. The value returned is for the number of entries available for the 2 MB page size; 4 MB pages require two 2 MB entries, so the number of entries available for the 4 MB page size is one-half the returned value. |

### CPUID Fn8000_0005_EBX L1 TLB Identifiers

See CPUID Fn8000_0005_EAX.

| Bits | Description |
|---|---|
| 31:24 | **L1DTlb4KAssoc: data TLB associativity for 4 KB pages**. Value: 0FFh. |
| 23:16 | **L1DTlb4KSize: data TLB number of entries for 4 KB pages**. Value: 48. |
| 15:8 | **L1ITlb4KAssoc: instruction TLB associativity for 4 KB pages**. Value: 0FFh. |
| 7:0 | **L1ITlb4KSize: instruction TLB number of entries for 4 KB pages**. Value: 32. |

### CPUID Fn8000_0005_ECX L1 Cache Identifiers

See CPUID Fn8000_0005_EAX.

| Bits | Description |
|---|---|
| 31:24 | **L1DcSize: L1 data cache size in KB**. Value: 64. |
| 23:16 | **L1DcAssoc: L1 data cache associativity**. Value: 2. |
| 15:8 | **L1DcLinesPerTag: L1 data cache lines per tag**. Value: 1. |
| 7:0 | **L1DcLineSize: L1 data cache line size in bytes**. Value: 64. |

### CPUID Fn8000_0005_EDX L1 Cache Identifiers

See CPUID Fn8000_0005_EAX.

| Bits | Description |
|---|---|
| 31:24 | **L1IcSize: L1 instruction cache size KB**. Value: 64. |
| 23:16 | **L1IcAssoc: L1 instruction cache associativity**. Value: 2. |
| 15:8 | **L1IcLinesPerTag.: L1 instruction cache lines per tag**. Value: 1. |
| 7:0 | **L1IcLineSize: L1 instruction cache line size in bytes**. Value: 64. |

### CPUID Fn8000_0006_EAX L2 Cache and L2 TLB Identifiers

This provides the processor's second level cache and TLB characteristics for each core.

The *associativity* fields are encoded as follows:

0h: The L2 cache or TLB is disabled.
1h: Direct mapped.
Ah: 32-way associative.
Bh: 48-way associative.

2h: 2-way associative.
4h: 4-way associative.
6h: 8-way associative.
8h: 16-way associative.
Ch: 64-way associative.
Dh: 96-way associative.
Eh: 128-way associative.
Fh: Fully associative.
All other encodings are reserved.

| Bits | Description |
|---|---|
| 31:28 | **L2DTlb2and4MAssoc: L2 data TLB associativity for 2 MB and 4 MB pages**. Value: 2. |
| 27:16 | **L2DTlb2and4MSize: L2 data TLB number of entries for 2 MB and 4 MB pages**. Value: 128. The value returned is for the number of entries available for the 2 MB page size; 4 MB pages require two 2 MB entries, so the number of entries available for the 4 MB page size is one-half the returned value. |
| 15:12 | **L2ITlb2and4MAssoc: L2 instruction TLB associativity for 2 MB and 4 MB pages**. Value: 0. |
| 11:0 | **L2ITlb2and4MSize: L2 instruction TLB number of entries for 2 MB and 4 MB pages**. Value: 0. |

### CPUID Fn8000_0006_EBX L2 Cache and L2 TLB Identifiers

See CPUID Fn8000_0006_EAX.

| Bits | Description |
|---|---|
| 31:28 | **L2DTlb4KAssoc: L2 data TLB associativity for 4 KB pages**. Value: 4. |
| 27:16 | **L2DTlb4KSize: L2 data TLB number of entries for 4 KB pages**. Value: 1024. |
| 15:12 | **L2ITlb4KAssoc: L2 instruction TLB associativity for 4 KB pages**. Value: 4. |
| 11:0 | **L2ITlb4KSize: L2 instruction TLB number of entries for 4 KB pages**. Value: 512. |

### CPUID Fn8000_0006_ECX L2 Cache and L2 TLB Identifiers

See CPUID Fn8000_0006_EAX.

| Bits | Description |
|---|---|
| 31:16 | **L2Size: L2 cache size in KB**. Value: Product-specific. May be one of 512 or 1024. |
| 15:12 | **L2Assoc: L2 cache associativity**. Value: 8. |
| 11:8 | **L2LinesPerTag: L2 cache lines per tag**. Value: 1. |
| 7:0 | **L2LineSize: L2 cache line size in bytes**. Value: 64. |

### CPUID Fn8000_0006_EDX Reserved

| Bits | Description |
|---|---|
| 31:0 | Reserved. |

### CPUID Fn8000_0007_E[A,B,C]X Advanced Power Management Information

| Bits | Description |
|---|---|
| 31:0 | Reserved. |

### CPUID Fn8000_0007_EDX Advanced Power Management Information

This function provides advanced power management feature identifiers.

| Bits | Description |
|---|---|
| 31:11 | Reserved. |
| 10 | **EffFreqRO: read-only effective frequency interface**. Value: 0. 1=Indicates presence of MSR0000_00E7 [Max Performance Frequency Clock Count (MPERF)] and MSR0000_00E8 [Actual Performance Frequency Clock Count (APERF)]. |
| 9 | **CPB: core performance boost is supported**. Value: Product-specific. |
| 8 | **TscInvariant: TSC invariant**. Value: 1. The TSC rate is invariant. |
| 7 | **HwPstate: hardware P-state control is supported**. Value: 1. [The P-State Current Limit Register] MSRC001_0061, [The P-State Control Register] MSRC001_0062 and [The P-State Status Register] MSRC001_0063 exist. |
| 6 | **100MHzSteps: 100 MHz multiplier Control**. Value: 1. |
| 5 | Reserved. |
| 4 | **TM: hardware thermal control (HTC) is supported**. Value: Product-specific. |
| 3 | **TTP: THERMTRIP is supported**. Value: 1. |
| 2 | **VID: Voltage ID control is supported**. Value: 0. Function replaced by HwPstate. |
| 1 | **FID: Frequency ID control is supported**. Value: 0. Function replaced by HwPstate. |
| 0 | **TS: Temperature sensor**. Value: 1. |

### CPUID Fn8000_0008_EAX Long Mode Address Size Identifiers

This provides information about the number of physical cores and the maximum physical and linear address width supported by the processor.

| Bits | Description |
|---|---|
| 31:16 | Reserved. |
| 15:8 | **LinAddrSize: Maximum linear byte address size in bits**. IF (CPUID Fn8000_0001_EDX[LM]==1) THEN Value: 30h. ELSE Value: 20h. ENDIF. |
| 7:0 | **PhysAddrSize: Maximum physical byte address size in bits**. Value: 28h. |

### CPUID Fn8000_0008_EBX Long Mode Address Size Identifiers

| Bits | Description |
|---|---|
| 31:0 | Reserved. |

### CPUID Fn8000_0008_ECX Long Mode Address Size Identifiers

This provides information about the number of physical cores and the maximum physical and linear address

width supported by the processor.

| Bits | Description |
|---|---|
| 31:16 | Reserved. |
| 15:12 | **ApicIdCoreIdSize: APIC ID size**. Value: 2h. The number of bits in the initial APIC20[ApicId] that indicate core ID within the processor. |
| 11:8 | Reserved. |
| 7:0 | **NC: number of physical cores - 1**. Value: Product-specific. The number of cores in the processor is NC+1 (e.g., if NC=0, then there is one core). See 2.4.2 [Processor Cores and Downcoring]. |

### CPUID Fn8000_0008_EDX Long Mode Address Size Identifiers

| Bits | Description |
|---|---|
| 31:0 | Reserved. |

### CPUID Fn8000_0009 Reserved

| Bits | Description |
|---|---|
| 31:0 | Reserved. |

IF (CPUID Fn8000_0001_ECX[SVM]==1) THEN.

### CPUID Fn8000_000A_EAX SVM Revision and Feature Identification

Provides SVM revision and feature information.

| Bits | Description |
|---|---|
| 31:8 | Reserved. |
| 7:0 | **SvmRev: SVM revision**. Value: 01h. |

ENDIF.

IF (CPUID Fn8000_0001_ECX[SVM]==1) THEN.

### CPUID Fn8000_000A_EBX SVM Revision and Feature Identification

Provides SVM revision and feature information.

| Bits | Description |
|---|---|
| 31:0 | **NASID: number of address space identifiers (ASID)**. Value: 40h. |

ENDIF.

### CPUID Fn8000_000A_ECX SVM Revision and Feature Identification

| Bits | Description |
|---|---|
| 31:0 | Reserved. |

IF (CPUID Fn8000_0001_ECX[SVM]==1) THEN.

**CPUID Fn8000_000A_EDX SVM Revision and Feature Identification**

Provides SVM revision and feature information.

| Bits | Description |
|---|---|
| 31:11 | Reserved. |
| 10 | **PauseFilter: pause intercept filter**. Value: 1. |
| 9:8 | Reserved. |
| 7 | **DecodeAssists: decode assists**. Value: 0. |
| 6 | **FlushByAsid: flush by ASID**. Value: 0. |
| 5 | **VmcbClean: VMCB clean bits**. Value: 0. |
| 4 | **TscRatio: TSC ratio**. Value: 0. |
| 3 | **NRIPS: NRIP save**. Value: 1. |
| 2 | **SVML: SVM lock**. Value: 1. |
| 1 | **LbrVirt: LBR virtualization**. Value: 1. |
| 0 | **NP: nested paging**. Value: 1. |

ENDIF.

**CPUID Fn8000_00[18:0B] Reserved**

| Bits | Description |
|---|---|
| 31:0 | Reserved. |

**CPUID Fn8000_0019_EAX TLB 1GB Page Identifiers**

This provides 1 GB paging information. The *associativity* fields are defined by CPUID Fn8000_0006_EAX, CPUID Fn8000_0006_EBX, CPUID Fn8000_0006_ECX and CPUID Fn8000_0006_EDX.

| Bits | Description |
|---|---|
| 31:28 | **L1DTlb1GAssoc: L1 data TLB associativity for 1 GB pages**. Value: 0Fh. |
| 27:16 | **L1DTlb1GSize: L1 data TLB number of entries for 1 GB pages**. Value: 48. |
| 15:12 | **L1ITlb1GAssoc: L1 instruction TLB associativity for 1 GB pages**. Value: 0. |
| 11:0 | **L1ITlb1GSize: L1 instruction TLB number of entries for 1 GB pages**. Value: 0. |

**CPUID Fn8000_0019_EBX TLB 1GB Page Identifiers**

This provides 1 GB paging information. The *associativity* fields are defined by CPUID Fn8000_0006_EAX, CPUID Fn8000_0006_EBX, CPUID Fn8000_0006_ECX and CPUID Fn8000_0006_EDX.

| Bits | Description |
|---|---|
| 31:28 | **L2DTlb1GAssoc: L2 data TLB associativity for 1 GB pages**. Value: 6. |
| 27:16 | **L2DTlb1GSize: L2 data TLB number of entries for 1 GB pages**. Value: 16. |

| Bits | Description |
|---|---|
| 15:12 | **L2ITlb1GAssoc: L2 instruction TLB associativity for 1 GB pages**. Value: 0. |
| 11:0 | **L2ITlb1GSize: L2 instruction TLB number of entries for 1 GB pages**. Value: 0. |

### CPUID Fn8000_0019_E[C,D]X TLB 1GB Page Identifiers

| Bits | Description |
|---|---|
| 31:0 | Reserved. |

### CPUID Fn8000_001A_EAX Performance Optimization Identifiers

This function returns performance related information.

| Bits | Description |
|---|---|
| 31:2 | Reserved. |
| 1 | **MOVU: movu**. Value: Product-specific. |
| 0 | **FP128: fp128**. Value: Product-specific. |

### CPUID Fn8000_001A_E[B,C,D]X Performance Optimization Identifiers

| Bits | Description |
|---|---|
| 31:0 | Reserved. |

### CPUID Fn8000_001B_EAX Instruction Based Sampling Identifiers

This function returns IBS feature information.

| Bits | Description |
|---|---|
| 31:8 | Reserved. |
| 7 | **RipInvalidChk: invalid RIP indication supported**. Value: 1. |
| 6 | **OpCntExt: IbsOpCurCnt and IbsOpMaxCnt extend by 7 bits**. Value: 1. |
| 5 | **BrnTrgt: branch target address reporting supported**. Value: 1. |
| 4 | **OpCnt: op counting mode supported**. Value: 1. |
| 3 | **RdWrOpCnt: read write of op counter supported**. Value: 1. |
| 2 | **OpSam: IBS execution sampling supported**. Value: 1. |
| 1 | **FetchSam: IBS fetch sampling supported**. Value: 1. |
| 0 | **IBSFFV: IBS feature flags valid**. Value: 1. |

### CPUID Fn8000_001B_E[B,C,D]X Instruction Based Sampling Identifiers

| Bits | Description |
|---|---|
| 31:0 | Reserved. |

## 3.20 MSRs - MSR0000_xxxx

See 3.1 [Register Descriptions and Mnemonics] for a description of the register naming convention. MSRs are accessed through x86 WRMSR and RDMSR instructions.

### MSR0000_0000 Load-Store MCA Address Register

| Bits | Description |
|---|---|
| 63:0 | Alias of MSR0000_040E. |

### MSR0000_0001 Load-Store MCA Status Register

| Bits | Description |
|---|---|
| 63:0 | Alias of MSR0000_040D. |

### MSR0000_0010 Time Stamp Counter Register (TSC)

Reset: 0000_0000_0000_0000h.

| Bits | Description |
|---|---|
| 63:0 | **TSC: time stamp counter**. Read-write. After reset, this register increments as defined by MSRC001_0015[TscFreqSel]. The TSC counts at the same rate in all P-states, all C states, and S0. |

### MSR0000_001B APIC Base Address Register (APIC_BAR)

| Bits | Description |
|---|---|
| 63:40 | MBZ. |
| 39:12 | **ApicBar: APIC base address register**. Read-write. Reset: 00FE_E00h. Specifies the base address for the APICXX register set. See 3.18 [APIC Registers] for details about this register set. |
| 11 | **ApicEn: APIC enable**. Read-write. Reset: 0b. 1=Local APIC enabled; the APICXX register set is accessible; all interrupt types are accepted. 0=Local APIC disabled; the APICXX register set is not accessible; only non-vectored interrupts are supported including NMI, SMI, INIT and ExtINT; local-vector-table interrupts can still occur if the LVTs have been previously programmed. |
| 10:9 | MBZ. |
| 8 | **BSC: boot strap core**. Read-write. Reset: xb. 1=The core is the BSC. 0=The core is not the BSC. |
| 7:0 | MBZ. |

### MSR0000_002A Cluster ID Register (EBL_CR_POWERON)

Reset: 0000_0000_0000_0000h. Attempted writes to this register result in general protection faults with error code 0.

| Bits | Description |
|---|---|
| 63:18 | MBZ. |

| | |
|---|---|
| 17:16 | **ClusterID**. Read-only. This is normally 00b; the value does not affect hardware. |
| 15:0 | MBZ. |

### MSR0000_00E7 Max Performance Frequency Clock Count (MPERF)

Reset: 0000_0000_0000_0000h.

| Bits | Description |
|---|---|
| 63:0 | **MPERF: maximum core clocks counter**. Read-write. This field is incremented by hardware at the P0 frequency while the core is in the C0 state. In combination with MSR0000_00E8, this is used to determine the effective frequency of the core. This field uses software P-state numbering. See MSRC001_0015[EffFreqCntMwait], 2.5.3.3 [Effective Frequency], and 2.5.3.1.2.1 [Software P-state Numbering]. |

### MSR0000_00E8 Actual Performance Frequency Clock Count (APERF)

Reset: 0000_0000_0000_0000h.

| Bits | Description |
|---|---|
| 63:0 | **APERF: actual core clocks counter**. Read-write. This field is incremented by hardware for each core clock cycle that occurs while the core is in the C0 state. In combination with MSR0000_00E7, this is used to determine the effective frequency of the core. See MSRC001_0015[EffFreqCntMwait] and 2.5.3.3 [Effective Frequency]. |

### MSR0000_00FE MTRR Capabilities Register (MTRRcap)

Reset: 0000_0000_0000_0508h.

| Bits | Description |
|---|---|
| 63:11 | Reserved. |
| 10 | **MtrrCapWc: write-combining memory type**. Read-only. 1=The write combining memory type is supported. |
| 9 | Reserved. |
| 8 | **MtrrCapFix: fixed range register**. Read-only. 1=Fixed MTRRs are supported. |
| 7:0 | **MtrrCapVCnt: variable range registers count**. Read-only. Specifies the number of variable MTRRs supported. |

### MSR0000_0174 SYSENTER CS Register (SYSENTER_CS)

Reset: 0000_0000_0000_0000h.

| Bits | Description |
|---|---|
| 63:32 | RAZ. |
| 31:16 | Reserved. |
| 15:0 | **SYSENTER_CS: SYSENTER target CS**. Read-write. Holds the called procedure code segment. |

### MSR0000_0175 SYSENTER ESP Register (SYSENTER_ESP)

Reset: 0000_0000_0000_0000h.

| Bits | Description |
|---|---|
| 63:32 | Reserved. |
| 31:0 | **SYSENTER_ESP: SYSENTER target SP**. Read-write. Holds the called procedure stack pointer. |

### MSR0000_0176 SYSENTER EIP Register (SYSENTER_EIP)

Reset: 0000_0000_0000_0000h.

| Bits | Description |
|---|---|
| 63:32 | Reserved. |
| 31:0 | **SYSENTER_EIP: SYSENTER target IP**. Read-write. Holds the called procedure instruction pointer. |

### MSR0000_0179 Global Machine Check Capabilities Register (MCG_CAP)

Reset: 0000_0000_0000_0106h.

| Bits | Description |
|---|---|
| 63:9 | Reserved. |
| 8 | **MCG_CTL_P: MCG_CTL register present**. Read-only. 1=The machine check control registers (MCi_CTL; see 2.16 [Machine Check Architecture]) are present. |
| 7:0 | **Count**. Read-only. Indicates the number of error reporting banks visible to each core. |

### MSR0000_017A Global Machine Check Status Register (MCG_STAT)

Reset: 0000_0000_0000_0000h. See 2.16 [Machine Check Architecture].

| Bits | Description |
|---|---|
| 63:3 | Reserved. |
| 2 | **MCIP: machine check in progress**. Read-write; set-by-hardware. 1=A machine check is in progress. |
| 1 | **EIPV: error instruction pointer valid**. Read-write; updated-by-hardware. 1=The instruction pointer that was pushed onto the stack by the machine check mechanism references the instruction that caused the machine check error. |
| 0 | **RIPV: restart instruction pointer valid**. Read-write; updated-by-hardware. 1=Program execution can be reliably restarted at the EIP address on the stack. 0=The interrupt was not precise and/or the process (task) context may be corrupt; continued operation of this process may not be possible without intervention, however system processing or other processes may be able to continue with appropriate software clean up. |

### MSR0000_017B Global Machine Check Exception Reporting Control Register (MCG_CTL)

Reset: 0000_0000_0000_0000h.

This register enables the various machine check register banks. See 2.16 [Machine Check Architecture]. It is expected that this register is programmed to the same value in all cores.
When a machine check register bank is disabled, errors for that bank are not detected or logged.

| Bits | Description |
|---|---|
| 63:6 | Reserved. |
| 5 | **MC5En: MC5 register bank enable**. Read-write. 1=The FR machine check register bank is enabled. |
| 4 | **MC4En: MC4 register bank enable**. Read-write. 1=The NB machine check register bank is enabled. |
| 3 | **MC3En: MC3 register bank enable**. Read-write. 1=The FP machine check register bank is enabled. |
| 2 | **MC2En: MC2 register bank enable**. Read-write. 1=The BU machine check register bank is enabled. |
| 1 | **MC1En: MC1 register bank enable**. Read-write. 1=The IF machine check register bank is enabled. |
| 0 | **MC0En: MC0 register bank enable**. Read-write. 1=The LS machine check register bank is enabled. |

### MSR0000_01D9 Debug Control Register (DBG_CTL_MSR)

Reset: 0000_0000_0000_0000h.

| Bits | Description |
|---|---|
| 63:7 | Reserved. |
| 6 | MBZ. |
| 5:2 | **PB: performance monitor pin control**. Read-write. This field does not control any hardware. |
| 1 | **BTF**. Read-write. 1=Enable branch single step. |
| 0 | **LBR**. Read-write. 1=Enable last branch record. |

### MSR0000_01DB Last Branch From IP Register (BR_FROM)

Reset: xxxx_xxxx_xxxx_xxxxh.

| Bits | Description |
|---|---|
| 63:0 | **LastBranchFromIP**. Read; GP-write. Returns the RIP of the last instruction before a control transfer.<br>Control Transfer — RIP Returned<br>Branch — RIP of the branch instruction.<br>Interrupt — RIP of the last instruction executed before the branch.<br>Exception — RIP of the instruction that caused the exception. |

### MSR0000_01DC Last Branch To IP Register (BR_TO)

Reset: xxxx_xxxx_xxxx_xxxxh.

| Bits | Description |
|---|---|
| 63:0 | **LastBranchToIP**. Read; GP-write. Returns the RIP of the target of the control transfer.<br>Control Transfer — RIP Returned<br>Branch — RIP of the branch target.<br>Interrupt — RIP of the interrupt handler.<br>Exception — RIP of the exception handler. |

### MSR0000_01DD Last Exception From IP Register

Reset: 0000_0000_0000_0000h.

| Bits | Description |
|---|---|
| 63:0 | **LastIntFromIP**. Read; GP-write. Holds the source RIP of the last control transfer that occurred before the exception or interrupt. |

### MSR0000_01DE Last Exception To IP Register

Reset: 0000_0000_0000_0000h.

| Bits | Description |
|---|---|
| 63:0 | **LastIntToIP**. Read; GP-write. Holds the target RIP of the last control transfer that occurred before the exception or interrupt. |

### MSR0000_020[E,C,A,8,6,4,2,0] Variable-Size MTRRs (MTRRphysBase*n*)

Reset: xxxx_xxxx_xxxx_xxxxh.

Each MTRR ([The Variable-Size MTRRs (MTRRphysBasen)] MSR0000_020[E,C,A,8,6,4,2,0], [The Fixed-Size MTRRs] MSR0000_02[6F:68,59,58,50], or [The MTRR Default Memory Type Register (MTRRdefType)] MSR0000_02FF) specifies a physical address range and a corresponding memory type (MemType) associated with that range. Setting MemType to an unsupported value results in a #GP(0).
The variable-size MTRRs come in pairs of base and mask registers (MSR0000_0200 and MSR0000_0201 are the first pair, etc.). Variables MTRRs are enabled through [The MTRR Default Memory Type Register (MTRRdefType)] MSR0000_02FF[MtrrDefTypeEn]. A CPU access--with address CPUAddr--is determined to be within the address range of a variable-size MTRR if the following equation is true:

CPUAddr[39:12] & PhyMask[39:12] == PhyBase[39:12] & PhyMask[39:12].

For example, if the variable MTRR spans 256K bytes and starts at the 1M byte address. The PhyBase would be set to 00_0010_0000h and the PhyMask to FF_FFFC_0000h (with zeros filling in for bits[11:0]). This results in a range from 00_0010_0000h to 00_0013_FFFFh.

| Bits | Description |
|---|---|
| 63:40 | MBZ. |

| | |
|---|---|
| 39:12 | **PhyBase: base address**. Read-write. |
| 11:3 | MBZ. |
| 2:0 | **MemType: memory type**. Read-write.<br>Bits Definition Bits Definition<br>0h UC or uncacheable. 5h WP or write protect.<br>1h WC or write combining. 6h WB or write back.<br>4h WT or write through.<br>All other values are reserved. |

### MSR0000_020[F,D,B,9,7,5,3,1] Variable-Size MTRRs (MTRRphysMask*n*)

Reset: xxxx_xxxx_xxxx_xxxxh. See MSR0000_020[E,C,A,8,6,4,2,0].

| Bits | Description |
|---|---|
| 63:40 | MBZ. |
| 39:12 | **PhyMask: address mask**. Read-write. |
| 11 | **Valid: valid**. Read-write. 1=The variable-size MTRR pair is enabled. |
| 10:0 | MBZ. |

### MSR0000_02[6F:68,59,58,50] Fixed-Size MTRRs

Reset: xxxx_xxxx_xxxx_xxxxh.

See MSR0000_020[E,C,A,8,6,4,2,0] for general MTRR information. Fixed MTRRs are enabled through MSR0000_02FF[MtrrDefTypeFixEn and MtrrDefTypeEn].

**Table 143: Fixed-size MTRR size and range mapping**

| Register | Bits | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 63:56 | 55:48 | 47:40 | 39:32 | 31:24 | 23:16 | 15:8 | 7:0 |
| MSR0000_0250 | 64K_70000 | 64K_60000 | 64K_50000 | 64K_40000 | 64K_30000 | 64K_20000 | 64K_10000 | 64K_00000 |
| MSR0000_0258 | 16K_9C000 | 16K_98000 | 16K_94000 | 16K_90000 | 16K_8C000 | 16K_88000 | 16K_84000 | 16K_80000 |
| MSR0000_0259 | 16K_BC000 | 16K_B8000 | 16K_B4000 | 16K_B0000 | 16K_AC000 | 16K_A8000 | 16K_A4000 | 16K_A0000 |
| MSR0000_0268 | 4K_C7000 | 4K_C6000 | 4K_C5000 | 4K_C4000 | 4K_C3000 | 4K_C2000 | 4K_C1000 | 4K_C0000 |
| MSR0000_0269 | 4K_CF000 | 4K_CE000 | 4K_CD000 | 4K_CC000 | 4K_CB000 | 4K_CA000 | 4K_C9000 | 4K_C8000 |
| MSR0000_026A | 4K_D7000 | 4K_D6000 | 4K_D5000 | 4K_D4000 | 4K_D3000 | 4K_D2000 | 4K_D1000 | 4K_D0000 |
| MSR0000_026B | 4K_DF000 | 4K_DE000 | 4K_DD000 | 4K_DC000 | 4K_DB000 | 4K_DA000 | 4K_D9000 | 4K_D8000 |
| MSR0000_026C | 4K_E7000 | 4K_E6000 | 4K_E5000 | 4K_E4000 | 4K_E3000 | 4K_E2000 | 4K_E1000 | 4K_E0000 |
| MSR0000_026D | 4K_EF000 | 4K_EE000 | 4K_ED000 | 4K_EC000 | 4K_EB000 | 4K_EA000 | 4K_E9000 | 4K_E8000 |
| MSR0000_026E | 4K_F7000 | 4K_F6000 | 4K_F5000 | 4K_F4000 | 4K_F3000 | 4K_F2000 | 4K_F1000 | 4K_F0000 |
| MSR0000_026F | 4K_FF000 | 4K_FE000 | 4K_FD000 | 4K_FC000 | 4K_FB000 | 4K_FA000 | 4K_F9000 | 4K_F8000 |

| Bits | Description |
|---|---|
| 63:61 | MBZ. |
| 60 | **RdDram: read DRAM**. Read-write. See: MSR0000_02[6F:68,59,58,50][4]. |
| 59 | **WrDram: write DRAM**. Read-write. See: MSR0000_02[6F:68,59,58,50][3]. |

| | |
|---|---|
| 58:56 | **MemType: memory type**. Read-write. See: MSR0000_02[6F:68,59,58,50][2:0]. |
| 55:53 | MBZ. |
| 52 | **RdDram: read DRAM**. Read-write. See: MSR0000_02[6F:68,59,58,50][4]. |
| 51 | **WrDram: write DRAM**. Read-write. See: MSR0000_02[6F:68,59,58,50][3]. |
| 50:48 | **MemType: memory type**. Read-write. See: MSR0000_02[6F:68,59,58,50][2:0]. |
| 47:45 | MBZ. |
| 44 | **RdDram: read DRAM**. Read-write. See: MSR0000_02[6F:68,59,58,50][4]. |
| 43 | **WrDram: write DRAM**. Read-write. See: MSR0000_02[6F:68,59,58,50][3]. |
| 42:40 | **MemType: memory type**. Read-write. See: MSR0000_02[6F:68,59,58,50][2:0]. |
| 39:37 | MBZ. |
| 36 | **RdDram: read DRAM**. Read-write. See: MSR0000_02[6F:68,59,58,50][4]. |
| 35 | **WrDram: write DRAM**. Read-write. See: MSR0000_02[6F:68,59,58,50][3]. |
| 34:32 | **MemType: memory type**. Read-write. See: MSR0000_02[6F:68,59,58,50][2:0]. |
| 31:29 | MBZ. |
| 28 | **RdDram: read DRAM**. Read-write. See: MSR0000_02[6F:68,59,58,50][4]. |
| 27 | **WrDram: write DRAM**. Read-write. See: MSR0000_02[6F:68,59,58,50][3]. |
| 26:24 | **MemType: memory type**. Read-write. See: MSR0000_02[6F:68,59,58,50][2:0]. |
| 23:21 | MBZ. |
| 20 | **RdDram: read DRAM**. Read-write. See: MSR0000_02[6F:68,59,58,50][4]. |
| 19 | **WrDram: write DRAM**. Read-write. See: MSR0000_02[6F:68,59,58,50][3]. |
| 18:16 | **MemType: memory type**. Read-write. See: MSR0000_02[6F:68,59,58,50][2:0]. |
| 15:13 | MBZ. |
| 12 | **RdDram: read DRAM**. Read-write. See: MSR0000_02[6F:68,59,58,50][4]. |
| 11 | **WrDram: write DRAM**. Read-write. See: MSR0000_02[6F:68,59,58,50][3]. |
| 10:8 | **MemType: memory type**. Read-write. See: MSR0000_02[6F:68,59,58,50][2:0]. |
| 7:5 | MBZ. |
| 4 | **RdDram: read DRAM**. IF (MSRC001_0010[MtrrFixDramModEn]== 1) THEN Read-write. ELSE MBZ. ENDIF. 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM. See MSRC001_0010[MtrrFixDramEn]. |
| 3 | **WrDram: write DRAM**. IF ( MSRC001_0010[MtrrFixDramModEn]== 1) THEN Read-write. ELSE MBZ. ENDIF. 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. See MSRC001_0010[MtrrFixDramEn]. |
| 2:0 | **MemType: memory type**. Read-write. |

| Bits | Definition | Bits | Definition |
|---|---|---|---|
| 0h | UC or uncacheable | 5h | WP or write protect |
| 1h | WC or write combining | 6h | WB or write back |
| 3h-2h | Reserved | 7h | Reserved |
| 4h | WT or write through. | | |

### MSR0000_0277 Page Attribute Table Register (PAT)

Reset: 0007_0406_0007_0406h. This register specifies the memory type based on the PAT, PCD, and PWT bits in the virtual address page tables. The encodings for PA[7:0] is:

0h = UC or uncacheable.
1h = WC or write combining.
4h = WT or write through.
All other values result in a #GP(0).
5h = WP or write protect.
6h = WB or write-back.
7h = UC- or uncacheable (overridden by MTRR WC state)

| Bits | Description |
|---|---|
| 63:59 | MBZ. |
| 58:56 | **PA7MemType**. Read-write. Default UC. MemType for {PAT, PCD, PWT} = 7h. |
| 55:51 | MBZ. |
| 50:48 | **PA6MemType**. Read-write. Default UC-. MemType for {PAT, PCD, PWT} = 6h. |
| 47:43 | MBZ. |
| 42:40 | **PA5MemType**. Read-write. Default WT. MemType for {PAT, PCD, PWT} = 5h. |
| 39:35 | MBZ. |
| 34:32 | **PA4MemType**. Read-write. Default WB. MemType for {PAT, PCD, PWT} = 4h. |
| 31:27 | MBZ. |
| 26:24 | **PA3MemType**. Read-write. Default UC. MemType for {PAT, PCD, PWT} = 3h. |
| 23:19 | MBZ. |
| 18:16 | **PA2MemType**. Read-write. Default UC-. MemType for {PAT, PCD, PWT} = 2h. |
| 15:11 | MBZ. |
| 10:8 | **PA1MemType**. Read-write. Default WT. MemType for {PAT, PCD, PWT} = 1h. |
| 7:3 | MBZ. |
| 2:0 | **PA0MemType**. Read-write. Default WB. MemType for {PAT, PCD, PWT} = 0h. |

### MSR0000_02FF MTRR Default Memory Type Register (MTRRdefType)

Reset: 0000_0000_0000_0000h. See MSR0000_020[E,C,A,8,6,4,2,0].

| Bits | Description |
|---|---|
| 63:12 | MBZ. |
| 11 | **MtrrDefTypeEn: variable and fixed MTRR enable**. Read-write. 1=[The Variable-Size MTRRs (MTRRphysBasen)] MSR0000_020[E,C,A,8,6,4,2,0], and [The Fixed-Size MTRRs] MSR0000_02[6F:68,59,58,50] are enabled. 0=Fixed and variable MTRRs are not enabled. |
| 10 | **MtrrDefTypeFixEn: fixed MTRR enable**. Read-write. 1=[The Fixed-Size MTRRs] MSR0000_02[6F:68,59,58,50], are enabled. This field is ignored (and the fixed MTRRs are not enabled) if MSR0000_02FF[MtrrDefTypeEn]=0. |

| | |
|---|---|
| 9:8 | MBZ. |
| 7:0 | **MemType: memory type**. Read-write. This specifies the memory type for memory space that is not specified by either the fixed or variable range MTRR's is defined as a function of MtrrDefTypeEn as follows:<br>• If MtrrDefTypeEn==1 then the default memory type is MemType.<br>• If MtrrDefTypeEn==0 then the default memory type is UC. |

### MSR0000_0400 DC Machine Check Control Register (MC0_CTL)

Reset: 0000_0000_0000_0000h. BIOS: 0000_0000_0000_00FFh. See 2.16 [Machine Check Architecture].

| Bits | Description |
|---|---|
| 63:8 | Unused. Read-only. |
| 7 | Unused. Read-write. |
| 6 | **L2TP: L2 TLB parity errors**. Read-write. 1=Report data cache L2 TLB parity errors. |
| 5 | **L1TP: L1 TLB parity errors**. Read-write. 1=Report data cache L1 TLB parity errors. |
| 4 | **DSTP: snoop tag array parity errors**. Read-write. 1=Report data cache snoop tag array parity errors. |
| 3 | **DMTP: main tag array parity errors**. Read-write. 1=Report data cache main tag array parity errors. |
| 2 | **DECC: data array ECC errors**. Read-write. 1=Report data cache data array ECC errors. If not set, ECC errors in the cache are detected and logged, but not reported. If masked (see MSRC001_0044), ECC errors in the cache are undetected. |
| 1 | **ECCM: multi-bit ECC data errors**. Read-write. 1=Report multi-bit ECC data errors during data cache line fills from the internal L2 or the system. If masked (see MSRC001_0044), multi-bit ECC errors on line fills may be detected and logged as single-bit errors unless single-bit ECC data errors are also masked (ECCI). If masking all line fill data errors is desired, all ECC data error mask bits (ECCI and ECCM) must be set. |
| 0 | **ECCI: single-bit ECC data errors**. Read-write. 1=Report single-bit ECC data errors during data cache line fills from the internal L2 or the system. If masked (see MSRC001_0044), multi-bit ECC errors on line fills may also be masked. If masking all line fill data errors is desired, all ECC data error mask bits (ECCI and ECCM) must be set. |

### MSR0000_0401 DC Machine Check Status Register (MC0_STATUS)

Cold reset: xxxx_xxxx_xxxx_xxxxh. See 2.16 [Machine Check Architecture]. Each of the MCi_STATUS registers hold information identifying the last error logged in each bank. Software is normally only allowed to write 0's to these registers to clear the fields so subsequent errors may be logged. See MSRC001_0015[McStatusWrEn]. The following field definitions apply to all MCi_STATUS registers, except as noted.

| Bits | Description |
|---|---|
| 63 | **Val: valid**. Read-write; set-by-hardware. 1=A valid error has been detected (whether it is enabled or not). This bit should be cleared to 0 by software after the register has been read. |
| 62 | **Over: error overflow**. Read-write; set-by-hardware. 1=An error was detected while the valid bit (Val) of this register was set; at least one error was not logged. The machine check mechanism handles the contents of MCi_STATUS during overflow as outlined in section 2.16.2.2 [Machine Check Error Logging Overwrite During Overflow]. |
| 61 | **UC: error uncorrected**. Read-write; updated-by-hardware. 1=The error was not corrected by hardware. |

| | |
|---|---|
| 60 | **En: error enable**. Read-write; updated-by-hardware. 1=MCA error reporting is enabled for this error in MCi_CTL. |
| 59 | **MiscV: miscellaneous error register valid**. Read-only. 1=MCi_MISC contains valid information for this error. This bit is always 0, except in the case of [The Reserved] MSR0000_0413 and [The FR Machine Check Miscellaneous Register (MC5_MISC)] MSR0000_0417. |
| 58 | **AddrV: error address valid**. Read-write; updated-by-hardware. 1=The address saved in MCi_ADDR is the address where the error occurred. |
| 57 | **PCC: processor context corrupt**. Read-write; updated-by-hardware. 1=The state of the processor may have been corrupted by the error condition. Restart may not be reliable. |
| 56:55 | Reserved. |
| 54:47 | **Syndrome[7:0]**. Read-write; updated-by-hardware.<br>• MC0_STATUS (DC): The lower eight syndrome bits when an ECC error is detected. See Table 50 for the mappings that show which bit errors result in which syndrome values.<br>• MC[3:1]_STATUS (LS, BU, IC): Reserved. |
| 46 | **CECC: correctable ECC error**. Read-write; updated-by-hardware. 1=The error was a correctable ECC error. |
| 45 | **UECC: uncorrectable ECC error**. Read-write; updated-by-hardware. 1=The error was an uncorrectable ECC error. |
| 44:32 | Reserved. |
| 31:24 | **Syndrome[15:8]**. Read-write; updated-by-hardware.<br>• MC0_STATUS (DC): The upper eight syndrome bits when an ECC error is detected. See Table 50 for the mappings that show which bit errors result in which syndrome values.<br>• MC[3:1]_STATUS (LS, BU, IC): Reserved. |
| 23:20 | Reserved. |
| 19:16 | **ErrorCodeExt: extended error code**. Read-write; updated-by-hardware. See the appropriate error signature tables below:<br>• MC0_STATUS (DC): Table 145<br>• MC1_STATUS (IC): Table 148<br>• MC2_STATUS (BU): Table 151<br>• MC3_STATUS (LS): Table 153<br>• MC5_STATUS (FR): Table 154 |
| 15:0 | **ErrorCode: error code**. Read-write; updated-by-hardware. See the appropriate error signature tables below:<br>• MC0_STATUS (DC): Table 145<br>• MC1_STATUS (IC): Table 148<br>• MC2_STATUS (BU): Table 151<br>• MC3_STATUS (LS): Table 153<br>• MC5_STATUS (FR): Table 154 |

This register reports these DC errors:

**Table 144: DC error descriptions**

| Error Type | Description | Enablers (MSR0000_0400 Control Bits) |
|---|---|---|
| L2 Cache Line Fill | An error occurred during an L1 line fill from the L2 cache. | ECC1, ECCM. |
| Data Load/ Store/ Victim/ Snoop | A data error occurred while accessing or managing data. | DECC |
| Tag Snoop/ Victim | A tag error was encountered during snoop or victimization. | DSTP |
| Tag Load/Store | A tag error was encountered during load or store. | |
| L1 TLB | Parity error in L1 TLB. | L1TP |
| L1 TLB Multi-match | Hit multiple entries. | L1TP |
| L2 TLB | Parity error in L2 TLB. | L2TP |
| L2 TLB Multi-match | Hit multiple entries. | L2TP |

**Table 145: DC error signatures**

| Error Type | [19:16] ErrorCodeExt | Error Code (see D18F3x48 for encoding) | | | | | | [61] UC | [58] ADDRV | [57] PCC | [54:47] Synd Valid | [46] CECC | [45] UECC |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Type | 10:9 PP | 8 T | 7:4 RRRR | 3:2 II/TT | 1:0 LL | | | | | | |
| L2 Cache Line Fill | 0000 | Mem | - | - | DRD | Data | L2 | If multi-bit | 1 | 1/0 | Y | If single-bit | If multi-bit |
| Data Load/ Store/ Victim/ Snoop | 0000 | Mem | - | - | DRD/ DWR/ Evict/ Snoop | Data | L1 | If multi-bit | 1/0 | 1/0 | Y | If single-bit | If multi-bit |
| Tag Snoop/ Victim | 0000 | Mem | - | - | Snoop/ Evict | Data | L1 | 1 | 1/0 | 1 | N | 0 | 0 |
| Tag Load/ Store | 0000 | Mem | - | - | DRD/ DWR | Data | L1 | 1 | 1 | 1 | N | 0 | 0 |
| L1 TLB | 0000 | TLB | - | - | - | Data | L1 | 1 | 1 | 1 | N | 0 | 0 |
| L1 TLB Multimatch | 0001 | TLB | - | - | - | Data | L1 | 1 | 1 | 1 | N | 0 | 0 |
| L2 TLB | 0000 | TLB | - | - | - | Data | L2 | 1 | 1 | 1 | N | 0 | 0 |
| L2 TLB Multimatch | 0001 | TLB | - | - | - | Data | L2 | 1 | 1 | 1 | N | 0 | 0 |

### MSR0000_0402 DC Machine Check Address Register (MC0_ADDR)

Cold reset: xxxx_xxxx_xxxx_xxxxh. See 2.16 [Machine Check Architecture]. This register contains valid data

if MSR0000_0401[AddrV]=1. Table 146 defines the address register as a function of error type.

| Bits | Description |
|---|---|
| 63:0 | **MC0_ADDR**. Read-write. See Table 146. |

**Table 146: DC error data; address register**

| Error Type | Memory Transaction Type (RRRR; Table 47) | Address Register Bits | Description |
|---|---|---|---|
| L2 Cache Line Fill | DRD | 47:6 | Physical address |
| Data Load/ Store/ Victim/ Snoop | DRD | 47:4[1] | Physical address |
| | DWR | | |
| | Evict | 11:6 | Physical address |
| | Snoop | | |
| Tag Snoop/ Victim | Snoop | 11:6 | Physical address |
| | Evict | | |
| Tag Load/ Store | DRD | 11:6[2] | Physical address |
| | DWR | | |
| L1 TLB | - | 47:12 | Linear address |
| L1 TLB Multi-match | | | |
| L2 TLB | | | |
| L2 TLB Multi-match | | | |

1. For Data Store (DWR), address bits shown are present only if error was reported (MSR0000_0401[UC] is set and MSR0000_0400[DECC] is enabled and not masked). If not reported, then valid address register bits are the linear address in 14:4.
2. The entire address from the TLB may be stored, but that address may only be incidentally related to the tag error; only the indicated bits are valid for this type of error.

### MSR0000_0403 DC Machine Check Miscellaneous Register (MC0_MISC)

Reset: 0000_0000_0000_0000h.

| Bits | Description |
|---|---|
| 63:0 | Reserved. |

### MSR0000_0404 IC Machine Check Control Register (MC1_CTL)

Reset: 0000_0000_0000_0000h. See 2.16 [Machine Check Architecture].

| Bits | Description |
|---|---|
| 63:32 | Unused. Read-only. |
| 31:10 | Unused. Read-write. |
| 9 | **RDDE: read data errors**. Read-write. 1-Report system read data errors for an instruction cache fetch if [The BU Machine Check Control Register (MC2_CTL)] MSR0000_0408[SRDE_ALL] = 1. |
| 8:7 | Unused. Read-write. |
| 6 | **L2TP: L2 TLB parity errors**. Read-write. 1=Report instruction cache L2 TLB parity errors. |
| 5 | **L1TP: L1 TLB parity errors**. Read-write. 1=Report instruction cache L1 TLB parity errors. |
| 4 | **ISTP: snoop tag array parity errors**. Read-write. 1=Report instruction cache snoop tag array parity errors. |
| 3 | **IMTP: main tag array parity errors**. Read-write. 1=Report instruction cache main tag array parity errors. |
| 2 | **IDP: data array parity errors**. Read-write. 1=Report instruction cache data array parity errors. |
| 1 | **ECCM: multi-bit ECC data errors**. Read-write. 1=Report multi-bit ECC data errors during instruction cache line fills or TLB reloads from the internal L2 or the system. |
| 0 | **ECCI: single-bit ECC data errors**. Read-write. 1=Report single-bit ECC data errors during instruction cache line fills or TLB reloads from the internal L2 or the system. |

### MSR0000_0405 IC Machine Check Status Register (MC1_STATUS)

Cold reset: xxxx_xxxx_xxxx_xxxxh. See 2.16 [Machine Check Architecture]. See MSRC001_0015[McStatusWrEn].

| Bits | Description |
|---|---|
| 63:0 | See: MSR0000_0401. |

**Table 147: IC error descriptions**

| Error Type | Description | Enablers (MSR0000_0404 Control Bits) |
|---|---|---|
| System Data Read Error | An error occurred during an attempted read of data from the NB. Possible reasons include master abort, target abort. | RDDE |
| L2 Cache Line Fill | An error occurred during a line fill from the L2 cache. | ECCM |
| IC Data Load (Parity) | A parity error occurred during load of data from the IC. This may be either a data error or a tag error. The data is discarded from the IC and can be refetched. | IDP, IMTP |
| Tag Snoop | A tag error was encountered during snoop or victimization. | ISTP |
| Copyback parity | A copyback parity error occurred. | IMTP |
| L1 TLB | Parity error in L1 TLB. | L1TP |

**Table 147: IC error descriptions**

| Error Type | Description | Enablers (MSR0000_0404 Control Bits) |
|---|---|---|
| L1 TLB Multi-match | Hit multiple entries. | L1TP |
| L2 TLB | Parity error in L2 TLB. | L2TP |
| L2 TLB Multi-match | Hit multiple entries. | L2TP |

**Table 148: IC error signatures**

| **Error Type** | **[19:16] ErrorCodeExt** | **Error Code (see D18F3x48 for encoding)** | | | | | | **[61] UC** | **[58] ADDRV** | **[57] PCC** | **[54:47] Synd Valid** | **[46] CECC** | **[45] UECC** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | **Type** | **10:9 PP** | **8 T** | **7:4 RRRR** | **3:2 II/TT** | **1:0 LL** | | | | | | |
| System Data Read Error | 0000 | BUS | SRC | 0 | IRD | MEM | LG | 1 | 0 | 0 | N | 0 | 0 |
| L2 Cache Line Fill | 0000 | Memory | - | - | IRD | Instr | L2 | 0[1] | 1 | 0 | N | 0[2] | 1 |
| IC Data Load (Parity) | 0000 | Memory | - | - | IRD | Instr | L1 | 0 | 1 | 0 | N | 0 | 0 |
| Tag Snoop | 0000 | Memory | - | - | Snoop | Instr | L1 | 1 | 1 | 1 | N | 0 | 0 |
| Copyback parity | 0000 | Memory | - | - | Evict | Instr | L1 | 0 | 0 | 0 | N | 0 | 0 |
| L1 TLB | 0000 | TLB | - | - | - | Instr | L1 | 0 | 1 | 0 | N | 0 | 0 |
| L1 TLB Multimatch | 0001 | TLB | - | - | - | Instr | L1 | 0 | 1 | 0 | N | 0 | 0 |
| L2 TLB | 0000 | TLB | - | - | - | Instr | L2 | 0 | 1 | 0 | N | 0 | 0 |
| L2 TLB Multimatch | 0001 | TLB | - | - | - | Instr | L2 | 0 | 1 | 0 | N | 0 | 0 |

1. Line refetched from memory. (Automatically purged from L2 during fill.)
2. Single bit errors are detected as parity errors.

### MSR0000_0406 IC Machine Check Address Register (MC1_ADDR)

Cold reset: xxxx_xxxx_xxxx_xxxxh. See 2.16 [Machine Check Architecture]. This register contains valid data if MSR0000_0405[AddrV]=1. Table 149 defines the address register as a function of error type.

| Bits | Description |
|---|---|
| 63:0 | **MC1_ADDR**. Read-write. See Table 149. |

**Table 149: IC error data; address register**

| Error Type | Address Register Bits | Description |
|---|---|---|
| L2 Cache Line Fill | 47:6 | Physical address |
| IC Data Load | 47:4 | Linear address |
| Tag Snoop | 47:6 | Physical address |
| L1 TLB | 47:12 for 4-Kbyte page<br>47:20 for 2-Mbyte page | Linear address |
| L1 TLB Multi-match | | |
| L2 TLB | 47:12 for 4-Kbyte page | Linear address |
| L2 TLB Multi-match | | |

### MSR0000_0407 IC Machine Check Miscellaneous Register (MC1_MISC)

| Bits | Description |
|---|---|
| 63:0 | Reserved. |

### MSR0000_0408 BU Machine Check Control Register (MC2_CTL)

Reset: 0000_0000_0000_0000h. See 2.16 [Machine Check Architecture].

| Bits | Description |
|---|---|
| 63:12 | Unused. Read-only. |
| 11 | **PDC_PAR: Pdc/GTLB parity errors**. Read-write. 1= Report Page Descriptor Cache parity or Guest TLB table walk parity errors. |
| 10 | **VB_PAR: write/victim data buffer parity error**. Read-write. 1=Report write buffer or victim buffer data parity errors. |
| 9 | Unused. Read-write. |
| 8 | **L2D_UECC: L2 data uncorrectable ECC error**. Read-write. 1=Report L2 data array uncorrectable ECC errors. |
| 7 | **L2D_CECC: L2 data correctable ECC error**. Read-write. 1=Report L2 data array correctable ECC errors. |
| 6 | **L2D_PAR: L2 data parity errors**. Read-write. 1=Report correctable and uncorrectable L2 data array parity errors. |
| 5 | **L2T_UECC: L2 tag uncorrectable ECC error**. Read-write. 1=Report L2 tag array uncorrectable ECC errors. |
| 4 | **L2T_CECC: L2 tag correctable ECC error**. Read-write. 1=Report L2 tag array correctable ECC errors. |
| 3 | **L2T_PAR: L2 tag parity errors**. Read-write. 1=Report L2 tag array correctable and uncorrectable parity errors. |
| 2 | **SRDE_ALL: all system read data**. Read-write. 1=Report system read data errors for any operation including a DC/IC fetch, TLB reload or hardware prefetch. |

| | |
|---|---|
| 1 | **SRDE_TLB: system read data TLB reload**. Read-write. 1=Report system read data errors for a TLB reload. |
| 0 | **SRDE_HP: system read data hardware prefetch**. Read-write. 1=Report system read data errors for a hardware prefetch. |

### MSR0000_0409 BU Machine Check Status Register (MC2_STATUS)

Cold reset: xxxx_xxxx_xxxx_xxxxh. See 2.16 [Machine Check Architecture]. See MSR0000_0401 for the information about this register. See MSRC001_0015[McStatusWrEn].

| Bits | Description |
|---|---|
| 63:0 | See: MSR0000_0401. |

**Table 150: BU error descriptions**

| Error Type | Description | Enablers (MSR0000_0408 Control Bits) |
|---|---|---|
| System Data Read Error | An error occurred during an attempted read of data from the NB. Possible reasons include master abort and target abort. | SRDE_ALL, SRDE_HP, SRDE_TLB |
| L2 Cache Data | A parity or ECC error occurred during a data access from the L2 cache. | L2D_CECC, L2D_UECC, L2D_PAR |
| Data Buffer | An error occurred in the write or victim data buffers. | VB_PAR |
| Data Copyback | An error occurred on a data copyback. | L2D_CECC, L2D_UECC, L2D_PAR |
| Tag | An error occurred in the L2 cache tags. | L2T_PAR, L2T_CECC, L2T_UECC |
| PDC/GTLB Parity | A parity error occurred in a PDC or GTLB. | PDC_PAR |

**Table 151: BU error signatures**

| **Error Type** | **Access Type** | **[19:16] ErrorCodeExt** | **Error Code (see D18F3x48 for encoding)** | | | | | | **[61] UC** | **[58] ADDRV** | **[57] PCC** | **[54:47] Synd Valid** | **[46] CECC** | **[45] UECC** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | **Type** | **10:9 PP** | **8 T** | **7:4 RRRR** | **3:2 II/TT** | **1:0 LL** | | | | | | |
| System Data Read Error | TLB | 0000 | BUS | SRC | 0 | RD | MEM/IO | LG | 1 | 1 | 0 | N | 0 | 0 |
| | HW Prefetch | 0000 | BUS | SRC | 0 | Prefetch | MEM/IO | LG | 1 | 0 | 0 | N | 0 | 0 |
| L2 Cache Data | TLB | 0000 | Mem | - | - | RD | Gen | L2 | 1/0 | 1 | 0 | N | 1/0 | 1/0 |
| Data buffer | Victim | 0011 | Mem | - | - | Snoop/Evict | Gen | LG | 1/0 | 1/0 | If UC | N | 0 | 0 |
| | Write | 0001 | Mem | - | - | WR | Gen | LG | 1/0 | 1/0 | If UC | N | 0 | 0 |
| Data Copyback | Snoop/Evict | 0000 | Mem | - | - | Snoop/Evict | Gen | L2 | 1/0 | 1 | If UC | N | 1/0 | 1/0 |

**Table 151: BU error signatures**

| Error Type | Access Type | [19:16] ErrorCodeExt | Error Code (see D18F3x48 for encoding) | | | | | | [61] UC | [58] ADDRV | [57] PCC | [54:47] Synd Valid | [46] CECC | [45] UECC |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Type | 10:9 PP | 8 T | 7:4 RRRR | 3:2 II/TT | 1:0 LL | | | | | | |
| Tag | Instr Fetch | 0010 | Mem | - | - | IRD | Instr | L2 | 1/0 | 1 | If UC | N | 1/0 | 1/0 |
| | Data Fetch | 0010 | Mem | - | - | DRD | Data | L2 | 1/0 | 1 | If UC | N | 1/0 | 1/0 |
| | TLB/ Snoop/ Evict | 0010 | Mem | - | - | RD/ Snoop/ Evict | Gen | L2 | 1/0 | 1 | If UC | N | 1/0 | 1/0 |
| PDC and GTLB Parity Error | Instr Fetch | 0000 | TLB | - | - | - | Instr | L1 | 1/0 | 1/0 | 1/0 | N | 0 | 0 |
| | Data Fetch | 0000 | TLB | - | - | - | Data | L1 | 1/0 | 1/0 | 1/0 | N | 0 | 0 |

### MSR0000_040A BU Machine Check Address Register (MC2_ADDR)

Cold reset: xxxx_xxxx_xxxx_xxxxh. See 2.16 [Machine Check Architecture]. This register contains valid data if MSR0000_0409[AddrV]=1. Table 152 defines the address register as a function of error type.

| Bits | Description |
|---|---|
| 63:0 | **MC2_ADDR**. Read-write. See Table 152. |

**Table 152: BU error data; address register**

| Error Type | Address Register Bits | Description |
|---|---|---|
| System Data Read Error | 47:6 | Physical address |
| L2 Cache Data | | |
| Data buffers | | |
| Data copyback | | |
| Tag | 3:0 | Encoded cache way |
| | 15:6 for 1-Mbyte L2<br>14:6 for 512-Kbyte L2<br>13:6 for 256-Kbyte L2<br>12:6 for 128-Kbyte L2 | Physical address |
| PDC/Guest TLB parity error | 47:2 | TLB reloader access or fetch address. Address bits 3:2 may not be valid for PDC errors. |

### MSR0000_040B BU Machine Check Miscellaneous Register (MC2_MISC)

Read-only. Reset: 0000_0000_0000_0000h.

| Bits | Description |
|---|---|
| 63:0 | Reserved. |

### MSR0000_040C LS Machine Check Control Register (MC3_CTL)

Reset: 0000_0000_0000_0000h. See 2.16 [Machine Check Architecture].

| Bits | Description |
|---|---|
| 63:2 | Unused. Read-only. |
| 1 | **SRDE_S: read data errors on store**. Read-write. 1=Report system read data errors on a store if [The BU Machine Check Control Register (MC2_CTL)] MSR0000_0408[SRDE_ALL] = 1. |
| 0 | **SRDE_L: read data errors on load**. Read-write. 1=Report system read data errors on a load if [The BU Machine Check Control Register (MC2_CTL)] MSR0000_0408[SRDE_ALL] = 1. |

### MSR0000_040D LS Machine Check Status Register (MC3_STATUS)

Cold reset: xxxx_xxxx_xxxx_xxxxh. See 2.16 [Machine Check Architecture]. See MSR0000_0401 for the information about all of the MCi_STATUS registers. See MSRC001_0015[McStatusWrEn].

| Bits | Description |
|---|---|
| 63:0 | See: MSR0000_0401. |

**Table 153: LS error signatures**

| Error Type | [19:16] ErrorCodeExt | Error Code (see D18F3x48 for encoding) | | | | | | [61] UC | [58] ADDRV | [57] PCC | [54:47] Synd Valid | [46] CECC | [45] UECC | [44:43] Reserved |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Type | 10:9 PP | 8 T | 7:4 RRRR | 3:2 II/TT | 1:0 LL | | | | | | | |
| Read Data on Store | 0000 | BUS | SRC | 0 | DWR | MEM | LG | 1 | 1/0 | 1/0 | N | 0 | 0 | |
| Read Data on Load | 0000 | BUS | SRC | 0 | DRD | MEM/IO | LG | 1 | 1/0 | 1/0 | N | 0 | 0 | |

### MSR0000_040E LS Machine Check Address Register (MC3_ADDR)

Cold reset: xxxx_xxxx_xxxx_xxxxh. See 2.16 [Machine Check Architecture]. This registers returns the physical address of the error if MSR0000_040D[AddrV]=1.

| Bits | Description |
|---|---|
| 63:0 | **MC3_ADDR**. Read-write. |

### MSR0000_040F LS Machine Check Miscellaneous Register (MC3_MISC)

Read-only. Reset: 0000_0000_0000_0000h.

| Bits | Description |
|---|---|
| 63:0 | Reserved. |

### MSR0000_0410 NB Machine Check Control Register (MC4_CTL)

Reset: xxxx_xxxx_xxxx_xxxxh. This register is also accessible through PCI configuration space. Only one of these registers exists in multi-core devices (see 3.1.1 [Northbridge MSRs In Multi-Core Products]).

| Bits | Description |
|---|---|
| 63:32 | Unused. Read-only. |
| 31:0 | Alias of D18F3x40. |

### MSR0000_0411 NB Machine Check Status Register (MC4_STATUS)

Reset: xxxx_xxxx_xxxx_xxxxh. See 2.16 [Machine Check Architecture] for an overview of the MCA registers. Table 117 lists and describes each error type logged. Table 118 and Table 119 describe the error codes and status register settings for each error type logged. Only one of these registers exists in multi-core devices (see 3.1.1 [Northbridge MSRs In Multi-Core Products]).

| Bits | Description |
|---|---|
| 63:32 | See: D18F3x4C. |
| 31:0 | See: D18F3x48. |

### MSR0000_0412 NB Machine Check Address Register (MC4_ADDR)

Cold reset: xxxx_xxxx_xxxx_xxxxh. Read-write. Only one of these registers exists in multi-core devices (see 3.1.1 [Northbridge MSRs In Multi-Core Products]).

| Bits | Description |
|---|---|
| 63:32 | See: D18F3x54. |
| 31:0 | See: D18F3x50. |

### MSR0000_0413 Reserved

Reset: 0000_0000_0000_0000h. Read-only.

| Bits | Description |
|---|---|
| 63:0 | Reserved. |

### MSR0000_0414 FR Machine Check Control Register (MC5_CTL)

Reset: 0000_0000_0000_0000h. See 2.16 [Machine Check Architecture].

| Bits | Description |
|---|---|
| 63:1 | Unused. Read-only. |
| 0 | **CPUWDT: CPU watchdog timer**. Read-write. 1=Enable core WDT expiration (see [The CPU Watchdog Timer Register (CpuWdtCfg)] MSRC001_0074). |

### MSR0000_0415 FR Machine Check Status Register (MC5_STATUS)

Cold reset: xxxx_xxxx_xxxx_xxxxh. See 2.16 [Machine Check Architecture]. See MSR0000_0401 for the information about all of the MCi_STATUS registers. See MSRC001_0015[McStatusWrEn].

| Bits | Description |
|---|---|
| 63:0 | See: MSR0000_0401. |

**Table 154: FR error signatures**

| Error Type | [19:16] Error-CodeExt | Error Code (see D18F3x48 for encoding) | | | | | | [61] UC | [58] ADD-RV | [57] PCC | [54:47] Synd Valid | [46] CECC | [45] UECC | [44] Reserved | [43] Reserved |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Type | 10:9 PP | 8 T | 7:4 RRRR | 3:2 II/TT | 1:0 LL | | | | | | | | |
| CPU watchdog timer expire | ---- | Bus | Gen | 1 | GEN | Gen | LG | 1 | 1 | 1 | No | 0 | 0 | 0 | 0 |

### MSR0000_0416 FR Machine Check Address Register (MC5_ADDR)

Cold reset: xxxx_xxxx_xxxx_xxxxh. See 2.16 [Machine Check Architecture]. This register returns the logical address of the next instruction after the last instruction retired if MSR0000_0415[AddrV]=1.

| Bits | Description |
|---|---|
| 63:0 | **MC5_ADDR**. Read-write. |

### MSR0000_0417 FR Machine Check Miscellaneous Register (MC5_MISC)

Cold reset: 0000_0000_0000_0xxxh. Read-only. This register records unspecified, implementation-specific status bits when an FR machine check error is logged.

| Bits | Description |
|---|---|
| 63:12 | Reserved. |
| 11:0 | **State**. Read-write. |

## 3.21 MSRs - MSRC000_0xxx

### MSRC000_0080 Extended Feature Enable Register (EFER)

Reset: 0000_0000_0000_0000h. SKINIT Execution: 0000_0000_0000_0000h.

| Bits | Description |
|---|---|
| 63:15 | MBZ. |
| 14 | **FFXSE: fast FXSAVE/FRSTOR enable**. Read-write. 1=Enables the fast FXSAVE/FRSTOR mechanism. A 64-bit operating system uses CPUID Fn0000_0001_EDX[24] to determine the presence of this feature before enabling it. This bit is set once by the operating system and its value is not changed afterwards. |
| 13 | **LMSLE: long mode segment limit enable**. Read-write. 1=Enables the long mode segment limit check mechanism. |
| 12 | **SVME: secure virtual machine (SVM) enable**. Read-write. 1=SVM features are enabled. |
| 11 | **NXE: no-execute page enable**. Read-write. 1=The no-execute page protection feature is enabled. |
| 10 | **LMA: long mode active**. Read-only. 1=Indicates that long mode is active. |
| 9 | MBZ. |
| 8 | **LME: long mode enable**. Read-write. 1=Long mode is enabled. |

| | |
|---|---|
| 7:1 | RAZ. |
| 0 | **SYSCALL: system call extension enable**. Read-write. 1=SYSCALL and SYSRET instructions are enabled. This adds the SYSCALL and SYSRET instructions which can be used in flat addressed operating systems as low latency system calls and returns. |

**MSRC000_0081 SYSCALL Target Address Register (STAR)**

Reset: 0000_0000_0000_0000h.

This register holds the target address used by the SYSCALL instruction and the code and stack segment selector bases used by the SYSCALL and SYSRET instructions.

| Bits | Description |
|---|---|
| 63:48 | **SysRetSel: SYSRET CS and SS**. Read-write. |
| 47:32 | **SysCallSel: SYSCALL CS and SS**. Read-write. |
| 31:0 | **Target: SYSCALL target address**. Read-write. |

**MSRC000_0082 Long Mode SYSCALL Target Address Register (STAR64)**

Reset: 0000_0000_0000_0000h.

| Bits | Description |
|---|---|
| 63:0 | **LSTAR: long mode target address**. Read-write. Target address for 64-bit mode calling programs. The address stored in this register must be in canonical form (if not canonical, a #GP fault occurs). |

**MSRC000_0083 Compatibility Mode SYSCALL Target Address Register (STARCOMPAT)**

Reset: 0000_0000_0000_0000h.

| Bits | Description |
|---|---|
| 63:0 | **CSTAR: compatibility mode target address**. Read-write. Target address for compatibility mode. The address stored in this register must be in canonical form (if not canonical, a #GP fault occurs). |

**MSRC000_0084 SYSCALL Flag Mask Register (SYSCALL_FLAG_MASK)**

Reset: 0000_0000_0000_0000h.

| Bits | Description |
|---|---|
| 63:32 | RAZ. |
| 31:0 | **MASK: SYSCALL flag mask**. Read-write. This register holds the EFLAGS mask used by the SYSCALL instruction. 1=Clear the corresponding EFLAGS bit when executing the SYSCALL instruction. |

**MSRC000_0100 FS Base Register (FS_BASE)**

Reset: 0000_0000_0000_0000h.

| Bits | Description |
|---|---|
| 63:0 | **FS_BASE: expanded FS segment base**. Read-write. This register provides access to the expanded 64-bit FS segment base. The address stored in this register must be in canonical form (if not canonical, a #GP fault fill occurs). |

**MSRC000_0101 GS Base Register (GS_BASE)**

Reset: 0000_0000_0000_0000h.

| Bits | Description |
|---|---|
| 63:0 | **GS_BASE: expanded GS segment base**. Read-write. This register provides access to the expanded 64-bit GS segment base. The address stored in this register must be in canonical form (if not canonical, a #GP fault fill occurs). |

**MSRC000_0102 Kernel GS Base Register (KernelGSbase)**

Reset: 0000_0000_0000_0000h.

| Bits | Description |
|---|---|
| 63:0 | **KernelGSBase: kernel data structure pointer**. Read-write. This register holds the kernel data structure pointer which can be swapped with the GS_BASE register using the SwapGS instruction. The address stored in this register must be in canonical form (if not canonical, a #GP fault occurs). |

**MSRC000_0103 Auxiliary Time Stamp Counter Register (TSC_AUX)**

Reset: 0000_0000_0000_0000h.

| Bits | Description |
|---|---|
| 63:32 | Reserved. |
| 31:0 | **TscAux: auxiliary time stamp counter data**. Read-write. It is expected that this is initialized by privileged software to a meaningful value, such as a processor ID. This value is returned in the RDTSCP instruction. |

## 3.22 MSRs - MSRC001_0xxx

**MSRC001_00[03:00] Performance Event Select Register (PERF_CTL[3:0])**

Reset: xxxx_xxxx_xxxx_xxxxh.

These registers are used to specify the events counted by the [The Performance Event Counter Registers (PERF_CTR[3:0])] MSRC001_00[07:04] and to control other aspects of their operation. Each performance counter supported has a corresponding event-select register that controls its operation. Section 3.24 [Performance Counter Events] shows the events and unit masks supported by the processor.

To accurately start counting with the write that enables the counter, disable the counter when changing the

event and then enable the counter with a second MSR write.

The edge count mode increments the counter when a transition happens on the monitored event. If the event selected is changed without disabling the counter, an extra edge is falsely detected when the first event is a static 0 and the second event is a static one. To avoid this false edge detection, disable the counter when changing the event and then enable the counter with a second MSR write.

The performance counter registers can be used to track events in the Northbridge. Northbridge events include all memory controller events and crossbar events documented in 3.24.7 and 3.24.8. Monitoring of Northbridge events should only be performed by one core. If a Northbridge event is selected using one of the Performance Event-Select registers in any core of a multi-core processor, then a Northbridge performance event cannot be selected in the same Performance Event Select register of any other core.

Care must be taken when measuring Northbridge or other non-processor-specific events under conditions where the processor may go into halt mode during the measurement period. For instance, one may wish to monitor DRAM traffic due to DMA activity from a disk or graphics adaptor. This entails running some event counter monitoring code on the processor, where such code accesses the counters at the beginning and end of the measurement period, or may even sample them periodically throughout the measurement period. Such code typically gives up the processor during each measurement interval. If there is nothing else for the OS to run on that particular processor at that time, it may halt the processor until it is needed. Under these circumstances, the clock for the counter logic may be stopped, hence the counters would not count the events of interest. To prevent this, simply run a low-priority background process that keeps the processor busy during the period of interest.

| Bits | Description |
|---|---|
| 63:42 | Reserved. |
| 41 | **HostOnly: host only counter**. Read-write. 1=Events are only counted when the processor is in host mode. |
| 40 | **GuestOnly: guest only counter**. Read-write. 1=Events are only counted when the processor is in guest mode. |
| 39:36 | Reserved. |
| 35:32 | **EventSelect[11:8]: performance event select**. Read-write. See: EventSelect[7:0]. |
| 31:24 | **CntMask: counter mask**. Read-write. Controls the number of events counted per clock cycle.<br>Bits — Definition<br>00h — The corresponding PERF_CTR[3:0] register increments by the number of events occurring in a clock cycle. Maximum number of events in one cycle is 32.<br>1Fh-01h — When Inv = 0, the corresponding PERF_CTR[3:0] register increments by 1, if the number of events occurring in a clock cycle is greater than or equal to the CntMask value.<br>When Inv = 1, the corresponding PERF_CTR[3:0] register increments by 1, if the number of events occurring in a clock cycle is less than CntMask value.<br>FFh-20h — Reserved |
| 23 | **Inv: invert counter mask**. Read-write. See CntMask. |
| 22 | **En: enable performance counter**. Read-write. 1= Performance event counter is enabled. |
| 21 | Reserved. |
| 20 | **Int: enable APIC interrupt**. Read-write. 1=APIC performance counter LVT interrupt is enabled to generate an interrupt when the performance counter overflows. |

| | |
|---|---|
| 19 | MBZ. |
| 18 | **Edge: edge detect**. Read-write. 0=Level detect. 1=Edge detect. |
| 17 | **OS: OS mode**. Read-write. 1=Events are only counted when CPL=0. |
| 16 | **User: user mode**. Read-write. 1=Events only counted when CPL>0. |
| 15:8 | **UnitMask: event qualification**. Read-write. Each UnitMask bit further specifies or qualifies the event specified by EventSelect. All events selected by UnitMask are simultaneously monitored. Unless otherwise stated, the UnitMask values shown may be combined (logically ORed) to select any desired combination of the sub-events for a given event. In some cases, certain combinations can result in misleading counts, or the UnitMask value is an ordinal rather than a bit mask. These situations are described where applicable, or should be obvious from the event descriptions. For events where no UnitMask table is shown, the UnitMask is not applicable and may be set to zeros. |
| 7:0 | **EventSelect[7:0]: event select**. Read-write. This field, along with EventSelect[11:8] above, combine to form the 12-bit event select field, EventSelect[11:0]. EventSelect specifies the event or event duration in a processor unit to be counted by the corresponding PERF_CTR[3:0] register. The events are specified in section 3.24 [Performance Counter Events]. Some events are reserved; when a reserved event is selected, the results are undefined. |

### MSRC001_00[07:04] Performance Event Counter Registers (PERF_CTR[3:0])

Reset: 0000_xxxx_xxxx_xxxxh.

Each core provides four 48-bit performance counters. Each counter can monitor a different event specified by [The Performance Event Select Register (PERF_CTL[3:0])] MSRC001_00[03:00]. The accuracy of the counters is not ensured.

Performance counters are used to count specific processor events, such as data-cache misses, or the duration of events, such as the number of clocks it takes to return data from memory after a cache miss. During event counting, the processor increments the counter when it detects an occurrence of the event. During duration measurement, the processor counts the number of processor clocks it takes to complete an event. Each performance counter can be used to count one event, or measure the duration of one event at a time.

In addition to the RDMSR instruction, the PERF_CTR[3:0] registers can be read using a special read performance-monitoring counter instruction, RDPMC. The RDPMC instruction loads the contents of the PERF_CTR[3:0] register specified by the ECX register, into the EDX register and the EAX register.

Writing the performance counters can be useful if there is an intention for software to count a specific number of events, and then trigger an interrupt when that count is reached. An interrupt can be triggered when a performance counter overflows. Software should use the WRMSR instruction to load the count as a two’s-complement negative number into the performance counter. This causes the counter to overflow after counting the appropriate number of times.

The performance counters are not assured of producing identical measurements each time they are used to measure a particular instruction sequence, and they should not be used to take measurements of very small instruction sequences. The RDPMC instruction is not serializing, and it can be executed out-of-order with respect to other instructions around it. Even when bound by serializing instructions, the system environment at the time the instruction is executed can cause events to be counted before the counter value is loaded into EDX:EAX.

| Bits | Description |
|---|---|
| 63:48 | RAZ. |
| 47:0 | **CTR: performance counter value**. Read-write. Returns the current value of the event counter. |

### MSRC001_0010 System Configuration Register (SYS_CFG)

Reset: 0000_0000_0002_0600h.

| Bits | Description |
|---|---|
| 63:23 | Reserved. |
| 22 | **Tom2ForceMemTypeWB: top of memory 2 memory type write-back**. Read-write. 1=The default memory type of memory between 4GB and TOM2 is write-back instead of the memory type defined by [The MTRR Default Memory Type Register (MTRRdefType)] MSR0000_02FF[MemType]. For this bit to have any effect, MSR0000_02FF[MtrrDefTypeEn] must be 1. MTRRs and PAT can be used to override this memory type. |
| 21 | **MtrrTom2En: MTRR top of memory 2 enable**. Read-write. 0=[The Top Of Memory 2 Register (TOM2)] MSRC001_001D is disabled. 1=This register is enabled. |
| 20 | **MtrrVarDramEn: MTRR variable DRAM enable**. Read-write. 0=[The Top Of Memory Register (TOP_MEM)] MSRC001_001A and IORRs are disabled. 1=These registers are enabled. This bit should be set by BIOS. |
| 19 | **MtrrFixDramModEn: MTRR fixed RdDram and WrDram modification enable**. Read-write. 0=MSR0000_02[6F:68,59,58,50][RdDram, WrDram] are MBZ. 1=MSR0000_02[6F:68,59,58,50][RdDram, WrDram] are read-write. This bit should be set to 1 during BIOS initialization of the fixed MTRRs, then cleared to 0 for operation. |
| 18 | **MtrrFixDramEn: MTRR fixed RdDram and WrDram attributes enable**. Read-write. BIOS: 1. 1=Enables the RdDram and WrDram attributes in MSR0000_02[6F:68,59,58,50]. |
| 17 | **SysUcLockEn: system lock command enable**. Read-write. 1=Transactions to the coherent fabric support the lock command. This is normally enabled in multi-core systems and disabled in single core systems. |
| 16 | **ChxToDirtyDis: change to dirty disable**. Read-write. 1=Disables change-to-dirty commands, evicts line from DC instead. |
| 15:11 | Reserved. |
| 10 | **SetDirtyEnO: clean-to-dirty command for O->M state transition enable**. Read-write. 1=Enables generating write probes when transitioning a cache line from Owned to Modified. |
| 9 | **SetDirtyEnS: shared-to-dirty command for S->M state transition enable**. Read-write. 1=Enables generating write probes when transitioning a cache line from Shared to Modified. |
| 8 | **SetDirtyEnE: shared-to-dirty command for E->M state transition enable**. Read-write. 1=Enables generating write probes when transitioning a cache line from Exclusive to Modified. |
| 7:0 | Reserved. |

**MSRC001_0015 Hardware Configuration Register (HWCR)**

Reset: 0000_0000_0000_0010h.

| Bits | Description |
|---|---|
| 63:27 | Reserved. |
| 26 | **EffFreqCntMwait: effective frequency counting during mwait**. Read-write. Specifies whether [The Max Performance Frequency Clock Count (MPERF)] MSR0000_00E7 and [The Actual Performance Frequency Clock Count (APERF)] MSR0000_00E8 increment while the core is in the monitor event pending state. 0=The registers do not increment. 1=The registers increment. See 2.5.3.3 [Effective Frequency]. |
| 25 | **CpbDis: core performance boost disable**. Read-write. Specifies whether core performance boost is enabled or disabled. 0=CPB is enabled. 1=CPB is disabled. See 2.5.3.1.1 [Core Performance Boost (CPB)]. If core performance boost is disabled while the core is in a boosted P-state, the core will automatically transition to the highest performance non-boosted P-state. |
| 24 | **TscFreqSel: TSC frequency select**. Read-write. Specifies the rate at which the TSC increments. 0=The TSC increments at the main PLL frequency (see D18F3xD4[MainPllOpFreqId]). 1=The TSC increments at the frequency defined by P0 at the time this bit is set by software. Changing the state of this bit after setting it to 1 results in undefined behavior from the TSC. Changing the CPU COFs defined by MSRC001_00[6B:64] after setting this bit has no effect on the TSC rate. BIOS should program this bit to a 1. This field uses software P-state numbering. See D18F4x15C[NumBoostStates] and 2.5.3.1.2.1 [Software P-state Numbering]. |
| 23 | **ForceUsRdWrSzPrb: force probes for upstream RdSized and WrSized**. Read-write. BIOS: See 2.9.3.5. 1=Forces probes on all upstream read-sized and write-sized transactions. This bit is shared between all processor cores. |
| 22 | Reserved. |
| 21 | **MisAlignSseDis: misaligned SSE mode disable**. Read-write. 1=Disables misaligned SSE mode. If this is set, then CPUID Fn8000_0001_ECX[MisAlignSse] is 0. |
| 20 | **IoCfgGpFault: IO-space configuration causes a GP fault**. Read-write. 1=IO-space accesses to configuration space cause a GP fault. The fault is triggered if any part of the IO read/write address range is between CF8h and CFFh, inclusive. These faults only result from single IO instructions, not to string and REP IO instructions. This fault takes priority over the IO trap mechanism described by [The IO Trap Control Register (SMI_ON_IO_TRAP_CTL_STS)] MSRC001_0054. |
| 19 | Reserved. |
| 18 | **McStatusWrEn: machine check status write enable**. Read-write. 1=Writes by software to MCi_STATUS (see 2.16 [Machine Check Architecture]) do not cause general protection faults; such writes update all implemented bits in these registers. To prevent undefined behavior, reserved fields must be written with 0. 0=Writing a non-zero pattern to these registers causes a general protection fault. |
| 17 | **Wrap32Dis: 32-bit address wrap disable**. Read-write. 1=Disable 32-bit address wrapping. Software can use Wrap32Dis to access physical memory above 4 Gbytes without switching into 64-bit mode. To do so, software should write a greater-than 4 Gbyte address to [The FS Base Register (FS_BASE)] MSRC000_0100 and [The GS Base Register (GS_BASE)] MSRC000_0101. Then it would address ±2 Gbytes from one of those bases using normal memory reference instructions with a FS or GS override prefix. However, the INVLPG, FST, and SSE store instructions generate 32-bit addresses in legacy mode, regardless of the state of Wrap32Dis. |

| | |
|---|---|
| 16 | Reserved. |
| 15 | **SseDis: SSE instructions disable**. Read-write. 1=Disables SSE instructions. If this is set, then CPUID Fn0000_0001_EDX[SSE, SSE2], CPUID Fn0000_0001_ECX[SSE3], and CPUID Fn8000_0001_ECX[SSE4A] are 0. |
| 14 | **RsmSpCycDis: RSM special bus cycle disable**. IF (SmmLock==0) THEN Read-write ELSE read-only. ENDIF. 0=A link special bus cycle, SMIACK, is generated on a resume from SMI. |
| 13 | **SmiSpCycDis: SMI special bus cycle disable**. IF (SmmLock==0) THEN Read-write ELSE read-only. ENDIF. 0=A link special bus cycle, SMIACK, is generated when an SMI interrupt is taken. |
| 12:11 | Reserved. |
| 10 | **MonMwaitUserEn: MONITOR/MWAIT user mode enable**. Read-write. 1=The MONITOR and MWAIT instructions are supported in all privilege levels. 0=The MONITOR and MWAIT instructions are supported only in privilege level 0; these instructions in privilege levels 1 to 3 cause a #UD exception. The state of this bit is ignored if MonMwaitDis is set. |
| 9 | **MonMwaitDis: MONITOR and MWAIT disable**. Read-write. 1=The MONITOR and MWAIT opcodes become invalid. This affects what is reported back through CPUID Fn0000_0001_ECX[Monitor]. |
| 8 | **IgnneEm: IGNNE port emulation enable**. Read-write. 1=Enable emulation of IGNNE port. |
| 7:6 | Reserved. |
| 5 | Reserved. |
| 4 | **INVDWBINVD: INVD to WBINVD conversion**. Read-write. 1=Convert INVD to WBINVD. BIOS: See 2.3.3 [Cache Initialization For General Storage During Boot]. |
| 3 | **TlbCacheDis: cacheable memory disable**. Read-write. 1=Disable performance improvement that assumes that the PML4, PDP, PDE and PTE entries are in cacheable WB DRAM memory. Operating systems that maintain page tables in any other memory type must set the TlbCacheDis bit to ensure proper operation. |
| 2 | Reserved. |
| 1 | **SlowFence: slow SFENCE enable**. Read-write. 1=Enable slow sfence. |
| 0 | **SmmLock: SMM code lock**. Read; write-1-only. SBIOS: 1. 1=SMM code in the ASeg and TSeg range and the SMM registers are read-only and SMI interrupts are not intercepted in SVM. |

### MSRC001_00[18,16] IO Range Registers Base (IORR_BASE[1:0])

MSRC001_0016 and MSRC001_0017 combine to specify the first IORR range and MSRC001_0018 and MSRC001_0019 combine to specify the second IORR range. A CPU access--with address CPUAddr--is determined to be within IORR address range if the following equation is true:

CPUAddr[39:12] & PhyMask[39:12] == PhyBase[39:12] & PhyMask[39:12].

| Bits | Description |
|---|---|
| 63:48 | RAZ. |
| 47:40 | MBZ. |
| 39:12 | **PhyBase: physical base address**. Read-write. Reset: X. |
| 11:5 | RAZ. |
| 4 | **RdMem: read from memory**. Read-write. Reset: X. 1=Read accesses to the range are directed to system memory. 0=Read accesses to the range are directed to IO. |

| | |
|---|---|
| 3 | **WrMem: write to memory**. Read-write. Reset: X. 1=Write accesses to the range are directed to system memory. 0=Write accesses to the range are directed to IO. |
| 2:0 | RAZ. |

### MSRC001_00[19,17] IO Range Registers Mask (IORR_MASK[1:0])

See MSRC001_00[18,16].

| Bits | Description |
|---|---|
| 63:48 | RAZ. |
| 47:40 | MBZ. |
| 39:12 | **PhyMask: physical address mask**. Read-write. Reset: X. |
| 11 | **Valid**. Read-write. Reset: X. 1=The pair of registers that specifies an IORR range is valid. |
| 10:0 | RAZ. |

### MSRC001_001A Top Of Memory Register (TOP_MEM)

| Bits | Description |
|---|---|
| 63:48 | RAZ. |
| 47:40 | MBZ. |
| 39:23 | **TOM[39:23]: top of memory**. Read-write. Reset: X. Specifies the address that divides between MMIO and DRAM. This value is normally placed below 4G. From TOM to 4G is MMIO; below TOM is DRAM. See 2.4.3 [Access Type Determination]. |
| 22:0 | RAZ. |

### MSRC001_001D Top Of Memory 2 Register (TOM2)

| Bits | Description |
|---|---|
| 63:48 | RAZ. |
| 47:40 | MBZ. |
| 39:23 | **TOM2[39:23]: second top of memory**. Read-write. Reset: X. Specifies the address divides between MMIO and DRAM. This value is normally placed above 4G. From 4G to TOM2 - 1 is DRAM; TOM2 and above is MMIO. See 2.4.3 [Access Type Determination]. This register is enabled by [The System Configuration Register (SYS_CFG)] MSRC001_0010[MtrrTom2En]. |
| 22:0 | RAZ. |

### MSRC001_001F Northbridge Configuration Register (NB_CFG)

Software is required to perform a read-modify-write in order to change any of the values in this register. Only one of these registers exists in multi-core devices; see 3.1.1 [Northbridge MSRs In Multi-Core Products].

| Bits | Description |
|---|---|
| 63:32 | Alias of D18F3x8C. |
| 31:0 | Alias of D18F3x88. |

### MSRC001_0022 Machine Check Exception Redirection Register

Reset: 0000_0000_0000_0000h.

This register can be used to redirect machine check exceptions (MCEs) to SMIs or vectored interrupts. If both RedirSmiEn and RedirVecEn are set, then undefined behavior results.

| Bits | Description |
|---|---|
| 63:10 | Reserved. |
| 9 | **RedirSmiEn**. Read-write. 1=Redirect MCEs (that are directed to this core) to generate an SMI-trigger IO cycle via MSRC001_0056. The status is stored in SMMFEC4[MceRedirSts]. |
| 8 | **RedirVecEn**. Read-write. 1=Redirect MCEs (that are directed to this core) to generate a vectored interrupt, using the interrupt vector specified in RedirVector. |
| 7:0 | **RedirVector**. Read-write. See RedirVecEn. |

### MSRC001_00[35:30] Processor Name String Registers

Reset: 0000_0000_0000_0000h.

These registers holds the CPUID name string in ASCII. The state of these registers are returned by CPUID instructions, CPUID Fn8000_000[4,3,2]_E[D,C,B,A]X. BIOS should set these registers to the product name for the processor as provided by AMD. Each register contains a block of 8 ASCII characters; the least byte corresponds to the first ASCII character of the block; the most-significant byte corresponds to the last character of the block. MSRC001_0030 contains the first block of the name string; MSRC001_0035 contains the last block of the name string.

| Bits | Description |
|---|---|
| 63:0 | **CpuNameString**. Read-write. |

### MSRC001_0044 DC Machine Check Control Mask (MC0_CTL_MASK)

Reset: 0000_0000_0000_0080h. BIOS: 0000_0000_0000_0080h.

| Bits | Description |
|---|---|
| 63:8 | Reserved. |
| 7 | Reserved. Read-write. |
| 6 | **L2TPMsk: L2 TLB parity error mask**. Read-write. Report data cache L2 TLB parity errors. |
| 5 | **L1TPMsk: L1 TLB parity error mask**.Read-write. |
| 4 | **DSTPMsk: snoop tag array parity error mask**. Read-write. |
| 3 | **DMTPMsk: main tag array parity error mask**. Read-write. R |
| 2 | **DECCMsk: data array ECC error mask**. Read-write. |
| 1 | **ECCMMsk: multi-bit ECC data error mask**. Read-write. |
| 0 | **ECCIMsk: single-bit ECC data error mask**. Read-write. |

### MSRC001_0045 IC Machine Check Control Mask (MC1_CTL_MASK)

Reset: 0000_0000_0000_0080h. BIOS: 0000_0000_0000_0080h.

| Bits | Description |
|---|---|
| 63:32 | Reserved. |
| 31:10 | Reserved. |
| 9 | **RDDEMsk: read data error mask**. Read-write. |
| 8:7 | Reserved. |
| 6 | **L2TPMsk: L2 TLB parity error mask**. Read-write. |
| 5 | **L1TPMsk: L1 TLB parity error mask**. Read-write. |
| 4 | **ISTPMsk: snoop tag array parity error mask**. Read-write. |
| 3 | **IMTPMsk: main tag array parity error mask**. Read-write. |
| 2 | **IDPMsk: data array parity error mask**. Read-write. |
| 1 | **ECCMMsk: multi-bit ECC data error mask**. Read-write. |
| 0 | **ECCIMsk: single-bit ECC data error mask**. Read-write. |

### MSRC001_0046 BU Machine Check Control Mask (MC2_CTL_MASK)

Reset: 0000_0000_0000_0200h. BIOS: 0000_0000_0000_0200h.

| Bits | Description |
|---|---|
| 63:12 | Reserved. |
| 11 | **PDC_PARMsk: Pdc/GTLB parity error mask**. Read-write. |
| 10 | **VB_PAR: write/victim data buffer parity error mask**. Read-write. |
| 9 | Reserved. |
| 8 | **L2D_UECCMsk: L2 data uncorrectable ECC error mask**. Read-write. |
| 7 | **L2D_CECCMsk: L2 data correctable ECC error mask**. Read-write. |
| 6 | **L2D_PARMsk: L2 data parity error mask**. Read-write. |
| 5 | **L2T_UECCMsk: L2 tag uncorrectable ECC error mask**. Read-write. |
| 4 | **L2T_CECCMsk: L2 tag correctable ECC error mask**. Read-write. |
| 3 | **L2T_PARMsk: L2 tag parity error mask**. Read-write. |
| 2 | **SRDE_ALLMsk: all system read data mask**. Read-write. |
| 1 | **SRDE_TLBMsk: system read data TLB reload mask**. Read-write. |
| 0 | **SRDE_HPMsk: system read data hardware prefetch mask**. Read-write. |

### MSRC001_0047 LS Machine Check Control Mask (MC3_CTL_MASK)

Reset: 0000_0000_0000_0000h. BIOS: 0000_0000_0000_0000h.

| Bits | Description |
|---|---|
| 63:2 | Reserved. |

| Bits | Description |
|---|---|
| 1 | **SRDE_SMsk: read data errors on store mask**. Read-write. |
| 0 | **SRDE_LMsk: read data errors on load mask**. Read-write. |

### MSRC001_0048 NB Machine Check Control Mask (MC4_CTL_MASK)

Reset: 0000_0000_0000_0000h. BIOS: 0000_0000_0000_0000h.

| Bits | Description |
|---|---|
| 63:32 | Reserved. |
| 31:26 | Reserved. |
| 25 | **McaUsPwDatErrMsk: MCA upstream data error mask**. Read-write. |
| 24:18 | Unused. |
| 17 | **DataMsk: data error reporting mask**. Read-write. |
| 16 | **ProtMsk: protocol error reporting mask**. Read-write. |
| 15:14 | Reserved. |
| 13 | **DevErrMsk: DEV error reporting mask**. Read-write. |
| 12 | **WDTRptMsk: watchdog timer error reporting mask**. Read-write. |
| 11 | **AtomicRMWMsk: atomic read-modify-write error reporting mask**. Read-write. |
| 10 | Reserved. |
| 9 | **TgtAbortMsk: target abort error reporting mask**. Read-write. |
| 8 | **MstrAbortMsk: master abort error reporting mask**. Read-write. |
| 7:6 | Reserved. |
| 5 | **SyncFloodMsk: sync flood error reporting mask**. Read-write. |
| 4:0 | Reserved. |

### MSRC001_0049 FR Machine Check Control Mask (MC5_CTL_MASK)

Reset: 0000_0000_0000_0000h.

| Bits | Description |
|---|---|
| 63:1 | Unused. Read-only. |
| 0 | **CPUWDTMsk: CPU watchdog timer mask**. Read-write. |

### MSRC001_00[53:50] IO Trap Registers (SMI_ON_IO_TRAP_[3:0])

Reset: 0000_0000_0000_0000h.

MSRC001_00[53:50] and MSRC001_0054 provide a mechanism for executing the SMI handler if a an access to one of the specified addresses is detected. Access address and access type checking is done before IO instruction execution. If the access address and access type match one of the specified IO address and access types, then: (1) the IO instruction is not executed; (2) any breakpoint, other than the single-step breakpoint, set on the IO instruction is not taken (the single-step breakpoint is taken after resuming from SMM); and (3) the

SMI-trigger IO cycle specified by MSRC001_0056. The status is stored in SMMFEC4[IoTrapSts].

IO-space configuration accesses are special IO accesses. An IO access is defined as an IO-space configuration access when IO instruction address bits[31:0] are CFCh, CFDh, CFEh, or CFFh when IO-space configuration is enabled (IOCF8[ConfigEn]). The access address for a configuration space access is the current value of IOCF8[BusNo, Device, Function, RegNo]. The access address for an IO access that is not a configuration access is equivalent to the IO instruction address, bits[31:0].

The access address is compared with SmiAddr, and the instruction access type is compared with the enabled access types defined by ConfigSMI, SmiOnRdEn, and SmiOnWrEn. Access address bits[23:0] can be masked with SmiMask.

IO and configuration space trapping to SMI applies only to single IO instructions; it does not apply to string and REP IO instructions.

| Bits | Description |
|---|---|
| 63 | **SmiOnRdEn: enable SMI on IO read**. Read-write. 1=Enables SMI generation on a read access. |
| 62 | **SmiOnWrEn: enable SMI on IO write**. Read-write. 1=Enables SMI generation on a write access. |
| 61 | **ConfigSmi: configuration space SMI**. Read-write. 1=Configuration access. 0=IO access (that is not an IO-space configuration access). |
| 60:56 | Reserved. |
| 55:32 | **SmiMask[23:0]**. Read-write. SMI IO trap mask. 0=Mask address bit. 1=Do not mask address bit. |
| 31:0 | **SmiAddr[31:0]**. Read-write. SMI IO trap address. |

### MSRC001_0054 IO Trap Control Register (SMI_ON_IO_TRAP_CTL_STS)

Reset: 0000_0000_0000_0000h.

For each of the SmiEn bits below, 1=The trap specified by the corresponding MSR is enabled. See MSRC001_00[53:50].

| Bits | Description |
|---|---|
| 63:32 | RAZ. |
| 31:16 | Reserved. |
| 15 | **IoTrapEn: IO trap enable**. Read-write. 1=Enable IO and configuration space trapping specified by MSRC001_00[53:50] and MSRC001_0054. |
| 14:8 | Reserved. |
| 7 | **SmiEn_3: SMI enable for the trap specified by MSRC001_0053**. Read-write. |
| 6 | Reserved. |
| 5 | **SmiEn_2: SMI enable for the trap specified by MSRC001_0052**. Read-write. |
| 4 | Reserved. |
| 3 | **SmiEn_1: SMI enable for the trap specified by MSRC001_0051**. Read-write. |
| 2 | Reserved. |
| 1 | **SmiEn_0: SMI enable for the trap specified by MSRC001_0050**. Read-write. |
| 0 | Reserved. |

### MSRC001_0055 Interrupt Pending

Reset: 0000_0000_0000_0000h.

| Bits | Description |
|---|---|
| 63:31 | Reserved. |
| 30 | **EnablePmTmrCheckLoop**. Read-write. 1=The core loops on IO-space read accesses to the address specified by IOMsgAddr until the data value has incremented from the previous read access. |
| 29:16 | Reserved. |
| 15:0 | **IOMsgAddr: IO message address**. Read-write. IO space message address. |

### MSRC001_0056 SMI Trigger IO Cycle Register

Reset: 0000_0000_0000_0000h.

See 2.4.6.2.3 [SMI Sources And Delivery]. This register specifies an IO cycle that may be generated when a local SMI trigger event occurs. If IoCycleEn is set and there is a local SMI trigger event, then the IO cycle generated is a byte read or write, based on IoRd, to address IoPortAddress. If the cycle is a write, then IoData contains the data written. If the cycle is a read, the value read is discarded. If IoCycleEn is clear and a local SMI trigger event occurs, then undefined behavior results.

| Bits | Description |
|---|---|
| 63:27 | Reserved. |
| 26 | **IoRd: IO Read**. Read-write. 1=IO read; 0=IO write. |
| 25 | **IoCycleEn: IO cycle enable**. Read-write. 1=The SMI trigger IO cycle is enabled to be generated. |
| 24 | Reserved. |
| 23:16 | **IoData**. Read-write. |
| 15:0 | **IoPortAddress**. Read-write. |

### MSRC001_0057 Reserved

Reset: 0000_0000_0000_0000h.

| Bits | Description |
|---|---|
| 63:0 | Reserved. |

### MSRC001_0058 MMIO Configuration Base Address

See 2.7 [Configuration Space] for a description of MMIO configuration space. All cores should program this register with the same value.

| Bits | Description |
|---|---|
| 63:48 | RAZ. |
| 47:40 | MBZ. |

<table>
<tr><td>39:20</td><td>MmioCfgBaseAddr[39:20]: MMIO configuration base address bits[39:20]. Read-write. Reset: X. Specifies the base address of the MMIO configuration range. The size of the MMIO configuration-space address range is defined by MSRC001_0058[BusRange] as follows. All lower order undefined bits must be 0.
<table>
<tr><th>BusRange</th><th>MmioCfgBaseAddr</th><th>BusRange</th><th>MmioCfgBaseAddr</th></tr>
<tr><td>0h</td><td>[39:20]</td><td>5h</td><td>[39:25]</td></tr>
<tr><td>1h</td><td>[39:21]</td><td>6h</td><td>[39:26]</td></tr>
<tr><td>2h</td><td>[39:22]</td><td>7h</td><td>[39:27]</td></tr>
<tr><td>3h</td><td>[39:23]</td><td>8h</td><td>[39:28]</td></tr>
<tr><td>4h</td><td>[39:24]</td><td>Fh-9h</td><td>Reserved</td></tr>
</table></td></tr>
<tr><td>19:6</td><td>RAZ.</td></tr>
<tr><td>5:2</td><td>BusRange: bus range identifier. Read-write. Reset: X. This specifies the number of busses in the MMIO configuration space range. The size of the MMIO configuration space range varies with this field as follows: the size is 1 Mbyte times the number of busses. This field is encoded as follows:
<table>
<tr><th>Bits</th><th>Busses</th><th>Bits</th><th>Busses</th></tr>
<tr><td>0h</td><td>1</td><td>5h</td><td>32</td></tr>
<tr><td>1h</td><td>2</td><td>6h</td><td>64</td></tr>
<tr><td>2h</td><td>4</td><td>7h</td><td>128</td></tr>
<tr><td>3h</td><td>8</td><td>8h</td><td>256</td></tr>
<tr><td>4h</td><td>16</td><td>Fh-9h</td><td>Reserved</td></tr>
</table></td></tr>
<tr><td>1</td><td>Reserved.</td></tr>
<tr><td>0</td><td>Enable. Read-write. Reset: 0. 1=MMIO configuration space is enabled.</td></tr>
</table>

### MSRC001_0060 BIST Results Register

Read-only. Reset: 0000_0000_xxxx_xxxxh.

This register provides BIST results after each reset. The results provided here are identical to the values provided in EAX. All 0’s indicates that no BIST failures were detected. 1=A failure was detected on that test.

| Bits | Description |
|---|---|
| 63:32 | Reserved. |
| 31:0 | **BistResults**. |

### MSRC001_0061 P-State Current Limit Register

See 2.5.3.1 [Core P-states]. Writes to this register cause a #GP.

| Bits | Description |
|---|---|
| 63:7 | RAZ. |
| 6:4 | **PstateMaxVal: P-state maximum value**. Read-only. Reset: Product-specific. Specifies the lowest-performance non-boosted P-state (highest non-boosted value) allowed. Attempts to change MSRC001_0062[PstateCmd] to a lower-performance P-state (higher value) are clipped to the value of this field. See 2.5.3.1.2.1 [Software P-state Numbering]. |
| 3 | RAZ. |
| 2:0 | **CurPstateLimit: current P-state limit**. Read-only. Reset: 0. Specifies the highest-performance non-boosted P-state (lowest value) allowed. MSRC001_0061[CurPstateLimit] is always bounded by MSRC001_0061[PstateMaxVal]. Attempts to change the CurPstateLimit to the value greater (lower performance) than MSRC001_0061[PstateMaxVal] leaves CurPstateLimit unchanged. See 2.5.3.1.2.1 [Software P-state Numbering].<br><br>CurPstateLimit = Max {000b, (D18F3x64[HtcPstateLimit] - D18F4x15C[NumBoostStates]) if D18F3x64[HtcAct]=1}.<br><br>The SB-TSI P-state limit register limits CurPstateLimit. See 2.10.2 [Sideband Temperature Sensor Interface (SB-TSI)]. |

### MSRC001_0062 P-State Control Register

| Bits | Description |
|---|---|
| 63:3 | MBZ. |
| 2:0 | **PstateCmd: P-state change command**. Read-write. Reset: Product-specific. Writes to this field cause the core to change to the indicated non-boosted P-state number, specified by MSRC001_00[6B:64]. 0=SWP0, 1=SWP1, etc. P-state limits are applied to any P-state requests made through this register. See 2.5.3.1 [Core P-states] and 2.5.3.1.2.1 [Software P-state Numbering]. Reads from this field return the last written value, regardless of whether any limits are applied. |

### MSRC001_0063 P-State Status Register

Writes to this register cause a #GP.

| Bits | Description |
|---|---|
| 63:3 | RAZ. |
| 2:0 | **CurPstate: current P-state**. Read-only; updated-by-hardware. Reset: values vary by product. Specifies the current non-boosted P-state of the core. 0=P0, 1=P1, etc. When a P-state change is requested, the value in this field is updated once all required frequency and voltage transitions are completed. See 2.5.3.1 [Core P-states] and 2.5.3.1.2.1 [Software P-state Numbering]. |

### MSRC001_00[6B:64] P-State [7:0] Registers

Reset: Product-specific. Per-node. All fields in these registers are required to be programmed to the same value

in each core. See 2.5.3.1 [Core P-states].

| Bits | Description |
|---|---|
| 63 | **PstateEn**. Read-write. 1=The P-state specified by this MSR is valid. 0=The P-state specified by this MSR is not valid. The purpose of this register is to indicate if the rest of the P-state information in the register is valid after a reset; it controls no hardware. MSRC001_00[6B:64][PstateEn] must be set in contiguous registers. |
| 62:42 | Reserved. |
| 41:40 | **IddDiv: current divisor field**. Read-write. After a reset, IddDiv and IddValue combine to specify the expected current dissipation of a single core that is in the P-state corresponding to the MSR number. These values are intended to be used in software algorithms, see 2.5.3.1.9.2 [_PSS (Performance Supported States)]), 2.5.3.1.8 [Processor-Systemboard Power Delivery Check]), and 2.5.1.4.1 [PSI_L Bit]. The values are expressed in amps; they are not intended to convey final product power levels; they may not match the power levels specified in the Power and Thermal Datasheets. These fields, may be subsequently altered by software; they do not affect the hardware behavior. These fields are encoded as follows:<br>Bits: 00b = IddValue / 1 A (0 to 255 A); 01b = IddValue / 10 A (0 to 25.5 A); 10b = IddValue / 100 A (0 to 2.55 A); 11b = Reserved |
| 39:32 | **IddValue: current value field**. Read-write. See MSRC001_00[6B:64][IddDiv]. |
| 31:16 | Reserved. |
| 15:9 | **CpuVid: CPU core VID**. Read-write. See 2.5.1 [Processor Power Planes And Voltage Control]. This field is required to be programmed as specified by MSRC001_0071[MaxVid, MinVid] otherwise undefined behavior results. |
| 8:4 | **CpuFid: CPU core frequency identifier**. Read-write. Specifies the frequency multiplier of the core.<br>• The CPU COF == 100MHz * (CpuFid + 10h) / (divisor specified by CpuDid). |
| 3:0 | **CpuDid: CPU core divisor identifier.** Read-write. Specifies the frequency divisor of the CPU.<br>0000b /1; 0001b /1.5; 0010b /2; 0011b /3; 0100b /4; 0101b /6; 0110b /8; 0111b /12; 1000b /16; 1111b-1001b reserved<br>See MSRC001_00[6B:64][CpuFid] for the CPU COF formula when in C0. |

### MSRC001_0071 COFVID Status Register

See 2.5.3.1 [Core P-states].

| Bits | Description |
|---|---|
| 63:59 | Reserved. |
| 58:56 | **CurPstateLimit: current P-state limit**. Read-only. Reset: 0. Provides the current lowest-performance P-state limit number. This register uses hardware P-state numbering. See MSRC001_0061[CurPstateLimit] and 2.5.3.1.2.2 [Hardware P-state Numbering]. |
| 55 | Reserved. |

| | |
|---|---|
| 54:49 | **MainPllOpFreqIdMax: main PLL operating frequency ID maximum**. Read-only. Reset: Product-specific. Specifies the maximum main PLL operating frequency supported by the processor. The maximum frequency is 100 MHz * (MainPllOpFreqIdMax + 10h), if MainPllOpFreqIdMax is greater than zero; if MainPllOpFreqIdMax = 00h, then there is no frequency limit. See D18F3xD4[MainPllOpFreqId]. |
| 48:42 | **MinVid: minimum voltage**. Read-only. Reset: Product-specific. Specifies the VID code corresponding to the minimum voltage (highest VID code) that the processor drives. 00h indicates that no minimum VID code is specified. See 2.5.1 [Processor Power Planes And Voltage Control]. |
| 41:35 | **MaxVid: maximum voltage**. Read-only. Reset: Product-specific. Specifies the VID code corresponding to the maximum voltage (lowest VID code) that the processor drives. 00h indicates that no maximum VID code is specified. See 2.5.1 [Processor Power Planes And Voltage Control]. |
| 34:32 | **StartupPstate: startup P-state number**. Read-only. Reset: Product-specific. Specifies the reset FID, DID, and VID for the core based on the P-state number selected. This field uses hardware P-state numbering. See MSRC001_00[6B:64]. |
| 31:25 | **CurNbVid: current northbridge VID**. Read-only. Reset: Product-specific. Specifies the current NB VID code request. |
| 24:21 | Reserved. |
| 20 | **PstateInProgress**. Read-only. Reset: 0. Specifies whether a core voltage or frequency transition is in progress. 1=Change is in progress. 0=No changes in progress. |
| 19 | Reserved. |
| 18:16 | **CurPstate: current P-state**. Read-only. Reset: Product-specific. Specifies the current P-state requested by the core. This field uses hardware P-state numbering. See MSRC001_0063[CurPstate] and 2.5.3.1.2.2 [Hardware P-state Numbering]. When a P-state change is requested, the value in this field is updated once all required frequency and voltage transitions are completed. |
| 15:9 | **CurCpuVid: current CPU core VID**. Read-only. Reset: Product-specific. Specifies the current VID code that has been sent to the voltage regulator. On a multi-core processor, CurCpuVid may not correspond to the VID code specified by the P-state currently requested by this core. When a P-state change is requested, the value in this field is updated after the voltage transition is complete, regardless of whether there is a following frequency transition. |
| 8:4 | **CurCpuFid: current CPU core frequency ID**. Read-only. Reset: Product-specific. Specifies the current CpuFid of the core. See MSRC001_00[6B:64]. When a P-state change is requested, the value in this field is updated once all required frequency and voltage transitions are completed. |
| 3:0 | **CurCpuDid: current CPU core divisor ID**. Read-only. Reset: Product-specific. Specifies the current CpuDid of the core. See MSRC001_00[6B:64]. When a P-state change is requested, the value in this field is updated once all required frequency and voltage transitions are completed. |

### MSRC001_0073 C-state Address Register

Reset: 0000_0000_0000_0000h.

| Bits | Description |
|---|---|
| 63:16 | Reserved. |
| 15:0 | **CstateAddr: C-state address**. Read-write. BIOS: 2.5.3.2.9. Specifies the IO addresses trapped by the core for C-state entry requests. A value of 0 in this field specifies that the core does not trap any IO addresses for C-state entry. Executing the HLT instruction still causes the core to enter a C-state. Writing values greater than 0FFF8h into this field results in undefined behavior. All other values cause the core to trap IO addresses CstateAddr through CstateAddr+7. See D18F4x118, D18F4x11C, and 2.5.3.2.2 [C-state Request Interface]. |

### MSRC001_0074 CPU Watchdog Timer Register (CpuWdtCfg)

Reset: 0000_0000_0000_0000h.

The CPU watchdog timer (WDT) is implemented as a counter that counts out the time periods specified. The counter starts counting when CpuWdtEn is set. The counter does not count during halt or stop-grant. It restarts the count each time an operation of an instruction completes. If no operation completes by the specified time period, then a machine check error may be recorded if enabled (see MSR0000_0414 through MSR0000_0417). If a watchdog timer error overflow occurs (MSR0000_0415[Overflow]), a sync flood can be generated if enabled in D18F3x180[SyncFloodOnCpuLeakErr].

The CPU watchdog timer must be set higher than the NB watchdog timer ([The MCA NB Configuration Register] D18F3x44) in order to allow remote requests to complete. The CPU watchdog timer must be set the same for all CPUs in a system

| Bits | Description |
|---|---|
| 63:7 | Reserved. |
| 6:3 | **CpuWdtCountSel: CPU watchdog timer count select**. Read-write. This, along with CpuWdtTimeBase, specifies the time period required for the WDT to expire. The time period is the value specified here times the time base specified by CpuWdtTimeBase. The actual timeout period may be anywhere from zero to one increments less than the values specified, due to non-deterministic behavior. The field is encoded as follows:<br>Bits — Definition<br>0000b — 4095<br>0001b — 2047<br>0010b — 1023<br>0011b — 511<br>0100b — 255<br>0101b — 127<br>0110b — 63<br>0111b — 31<br>1000b — 8191<br>1001b — 16383<br>1111b-1010b — Reserved |

| | |
|---|---|
| 2:1 | **CpuWdtTimeBase: CPU watchdog timer time base**. Read-write. Specifies the time base for the timeout period specified in CpuWdtCountSel.<br>Bits: Definition<br>00b: 1.31 millisecond<br>01b: 1.28 microsecond<br>10b: 80 nanoseconds<br>11b: Reserved |
| 0 | **CpuWdtEn: CPU watchdog timer enable**. Read-write. 1=The WDT is enabled. |

### MSRC001_0111 SMM Base Address Register (SMM_BASE)

Reset: 0000_0000_0003_0000h.

This holds the base of the SMM memory region. The value of this register is stored in the save state on entry into SMM (see 2.4.6.2.5 [SMM Save State]) and it is restored on returning from SMM. The 16-bit CS (code segment) selector is loaded with SMM_BASE[19:4] on entering SMM. SMM_BASE[3:0] must be 0.

If (SMM_BASE>=0010_0000h) then:
- The value of the CS selector is undefined upon SMM entry.
- The undefined CS selector value should not be used as the target of a far jump, call, or return.

The SMM base address can be changed in two ways:
- The SMM base address, at offset FF00h in the SMM state save area, may be changed by the SMI handler. The RSM instruction updates SMM_BASE with the new value.
- Normal WRMSR access to this register.

| Bits | Description |
|---|---|
| 63:32 | Reserved. |
| 31:0 | **SMM_BASE**. IF (MSRC001_0015[SmmLock]==1) THEN read-only ELSE Read-write ENDIF. |

### MSRC001_0112 SMM TSeg Base Address Register (SMMAddr)

Reset: 0000_0000_0000_0000h.

See 2.4.6.2 [System Management Mode (SMM)] for information about SMM. See MSRC001_0113.

Each CPU access, directed at CPUAddr, is determined to be in the TSeg range if the following is true:

CPUAddr[39:17] & TSegMask[39:17] == TSegBase[39:17] & TSegMask[39:17].

For example, if TSeg spans 256K bytes and starts at the 1M byte address. The MSRC001_0112[TSegBase] would be set to 0010_0000h and the MSRC001_0113[TSegMask] to FFFC_0000h (with zeros filling in for bits[16:0]). This results in a TSeg range from 0010_0000h to 0013_FFFFh.

| Bits | Description |
|---|---|
| 63:40 | Reserved. |
| 39:17 | **TSegBase[39:17]: TSeg address range base**. IF (MSRC001_0015[SmmLock]==1) THEN read-only ELSE Read-write ENDIF. |
| 16:0 | Reserved. |

### MSRC001_0113 SMM TSeg Mask Register (SMMMask)

Reset: 0000_0000_0000_0000h.

See 2.4.6.2 [System Management Mode (SMM)] for information about SMM.

The ASeg address range is located at a fixed address from A0000h–BFFFFh. The TSeg range is located at a variable base (specified by MSRC001_0112[TSegBase]) with a variable size (specified by MSRC001_0113[TSegMask]). These ranges provide a safe location for SMM code and data that is not readily accessible by non-SMM applications. The SMI handler can be located in one of these two ranges, or it can be located outside these ranges. These ranges must never overlap each other.

This register specifies how accesses to the ASeg and TSeg address ranges are control as follows:

- If [A, T]Valid=0, then the address range is accessed as specified by MTRRs, regardless of whether the CPU is in SMM or not.
- If [A, T]Valid=1, then:
  - If in SMM, then:
    - If [A, T]Close=0, then the accesses are directed to DRAM with memory type as specified in [A, T]MTypeDram.
    - If [A, T]Close=1, then instruction accesses are directed to DRAM with memory type as specified in [A, T]MTypeDram and data accesses are directed at MMIO space and with attributes based on [A, T]MTypeIoWc.
  - If not in SMM, then the accesses are directed at MMIO space with attributes based on [A, T]MTypeIoWc.

| Bits | Description |
|---|---|
| 63:40 | Reserved. |
| 39:17 | **TSegMask[39:17]: TSeg address range mask**. IF (MSRC001_0015[SmmLock]==1) THEN read-only ELSE Read-write ENDIF. See MSRC001_0112. |
| 16:15 | Reserved. |
| 14:12 | **TMTypeDram: TSeg address range memory type**. IF (MSRC001_0015[SmmLock]==1) THEN read-only ELSE Read-write ENDIF. Specifies the memory type for SMM accesses to the TSeg range that are directed to DRAM. The encoding is identical to the three LSBs of the MTRRs. See MSR0000_020[E,C,A,8,6,4,2,0]. |
| 11 | Reserved. |
| 10:8 | **AMTypeDram: ASeg Range Memory Type**. IF (MSRC001_0015[SmmLock]==1) THEN read-only ELSE Read-write ENDIF. Specifies the memory type for SMM accesses to the ASeg range that are directed to DRAM. The encoding is identical to the three LSBs of the MTRRs. See MSR0000_020[E,C,A,8,6,4,2,0]. |
| 7:6 | Reserved. |
| 5 | **TMTypeIoWc: non-SMM TSeg address range memory type**. IF (MSRC001_0015[SmmLock]==1) THEN read-only ELSE Read-write ENDIF. Specifies the attribute of TSeg accesses that are directed to MMIO space. 0=UC (uncacheable). 1=WC (write combining). |
| 4 | **AMTypeIoWc: non-SMM ASeg address range memory type**. IF (MSRC001_0015[SmmLock]==1) THEN read-only ELSE Read-write ENDIF. Specifies the attribute of ASeg accesses that are directed to MMIO space. 0=UC (uncacheable). 1=WC (write combining). |

| | |
|---|---|
| 3 | **TClose: send TSeg address range data accesses to MMIO**. Read-write. 1=When in SMM, direct data accesses in the TSeg address range to MMIO space. See, AClose. |
| 2 | **AClose: send ASeg address range data accesses to MMIO**. Read-write. 1=When in SMM, direct data accesses in the ASeg address range to MMIO space.<br><br>[A,T]Close allows the SMI handler to access the MMIO space located in the same address region as the [A,T]Seg. When the SMI handler is finished accessing the MMIO space, it must clear the bit. Failure to do so before resuming from SMM causes the CPU to erroneously read the save state from MMIO space. |
| 1 | **TValid: enable TSeg SMM address range**. IF (MSRC001_0015[SmmLock]==1) THEN read-only ELSE Read-write ENDIF. 1=The TSeg address range SMM enabled. |
| 0 | **AValid: enable ASeg SMM address range**. IF (MSRC001_0015[SmmLock]==1) THEN read-only ELSE Read-write ENDIF. 1=The ASeg address range SMM enabled. |

### MSRC001_0114 Virtual Machine Control Register (VM_CR)

Reset: 0000_0000_0000_0000h.

| Bits | Description |
|---|---|
| 63:32 | Reserved. |
| 31:5 | MBZ. |
| 4 | **Svme_Disable: SVME disable**. 1=MSRC000_0080[SVME] must be zero (MBZ) when writing to MSRC000_0080. Setting this bit when MSRC000_0080[SVME]=1 causes a #GP fault, regardless of the state of Lock. 0=MSRC000_0080[SVME] is read-write. See Lock. |
| 3 | **Lock: SVM lock**. Read-only; write-1-only; cleared-by-hardware. 1=Svme_Disable is read-only. 0=Svme_Disable is read-write. See MSRC001_0118. |
| 2 | **dis_a20m: disable A20 masking**. Read-write; set-by-hardware. 1=Disables A20 masking. This bit is set by hardware when the SKINIT instruction is executed. |
| 1 | **r_init: intercept INIT**. Read-write; set-by-hardware. This bit controls how INIT is delivered in host mode. This bit is set by hardware when the SKINIT instruction is executed.<br>0 = INIT delivered normally.<br>1 = INIT translated into a SX interrupt. |
| 0 | **dpd: debug port disable**. Read-write; set-by-hardware. This bit controls if debug facilities such as JTAG and HDT have access to the processor state information. This bit is set by hardware when the SKINIT instruction is executed.<br>0 = Debug port may be enabled.<br>1 = Debug port disabled; all mechanisms that could expose trusted code execution are disabled. |

### MSRC001_0115 IGNNE Register (IGNNE)

| Bits | Description |
|---|---|
| 63:32 | Reserved. |
| 31:1 | MBZ. |
| 0 | **IGNNE: current IGNNE state**. Read-write. Reset: X. This bit controls the current state of the processor internal IGNNE signal. |

### MSRC001_0116 SMM Control Register (SMM_CTL)

IF (MSRC001_0015[SmmLock]==1) THEN GP-read-write. ELSE GP-Read; write-only. ENDIF. The bits in this register are processed in the order of: smm_enter, smi_cycle, smm_dismiss, rsm_cycle and smm_exit. However, only the following combination of bits may be set in a single write (all other combinations result in undefined behavior):

- smm_enter and smi_cycle.
- smm_enter and smm_dismiss.
- smm_enter, smi_cycle and smm_dismiss.
- smm_exit and rsm_cycle.

Software is responsible for ensuring that smm_enter and smm_exit operations are properly matched and are not nested.

| Bits | Description |
|---|---|
| 63:5 | MBZ. |
| 4 | **rsm_cycle: send RSM special cycle**. 1=Send a RSM special cycle. |
| 3 | **smm_exit: exit SMM**.1=Exit SMM. |
| 2 | **smi_cycle: send SMI special cycle**. 1=Send a SMI special cycle. |
| 1 | **smm_enter: enter SMM**. 1=Enter SMM. |
| 0 | **smm_dismiss: clear SMI**. 1=Clear the SMI pending flag. |

### MSRC001_0117 Virtual Machine Host Save Physical Address Register (VM_HSAVE_PA)

Reset: 0000_0000_0000_0000h.

| Bits | Description |
|---|---|
| 63:40 | MBZ. |
| 39:12 | **VM_HSAVE_PA: physical address of host save area**. Read-write. This register contains the physical address where VMRUN saves host state and where vmexit restores host state from. Writing this register causes a #GP if the address written is >= FD_0000_0000h. |
| 11:0 | MBZ. |

### MSRC001_0118 SVM Lock Key

Reset: 0000_0000_0000_0000h.

| Bits | Description |
|---|---|
| 63:0 | **SvmLockKey: SVM lock key**. RAZ, write-only. Writes to this register when MSRC001_0114[Lock]=0 write the SvmLockKey. Writes to this register when MSRC001_0114[Lock]=1 and SvmLockKey!=0 cause hardware to clear MSRC001_0114[Lock] if the value written is the same as the value stored in SvmLockKey. |

#### MSRC001_0119 SMM Lock Key

Reset: 0000_0000_0000_0000h.

| Bits | Description |
|---|---|
| 63:0 | **SmmLockKey: SMM lock key**. RAZ, write-only. Writes to this register when MSRC001_0015[SmmLock]=0 write the SmmLockKey. Writes to this register when MSRC001_0015[SmmLock]=1 and SmmLockKey!=0 cause hardware to clear MSRC001_0015[SmmLock] if the value written is the same as the value stored in SmmLockKey. |

#### MSRC001_011A Local SMI Status

Reset: 0000_0000_0000_0000h. This registers returns the same information that is returned in [The Local SMI Status] SMMFEC4 portion of the SMM save state. The information in this register is only updated when MSRC001_0116[smm_dismiss] is set by software.

| Bits | Description |
|---|---|
| 63:32 | Reserved. |
| 31:0 | See: SMMFEC4 [Local SMI Status]. |

#### MSRC001_0140 OS Visible Work-around MSR0 (OSVW_ID_Length)

Reset: 0000_0000_0000_0000h.

| Bits | Description |
|---|---|
| 63:16 | Reserved. |
| 15:0 | **OSVW_ID_Length: OS visible work-around ID length**. Read-write. See the *Revision Guide for AMD Family 12h Processors* for the definition of this field. |

#### MSRC001_0141 OS Visible Work-around MSR1 (OSVW Status)

Reset: 0000_0000_0000_0000h.

| Bits | Description |
|---|---|
| 63:0 | **OsvwStatusBits: OS visible work-around status bits**. Read-write. See the *Revision Guide for AMD Family 12h Processors* for the definition of this field. |

### 3.23 MSRs - MSRC001_1xxx

#### MSRC001_1004 CPUID Features Register (Features)

MSRC001_1004 and MSRC001_1005 provide some control over values read from CPUID functions.

| Bits | Description |
|---|---|
| 63:32 | **FeaturesEcx**. Read-write. Provides back-door control over the features reported in CPUID function 1, ECX (see CPUID Fn0000_0001_ECX). |
| 31:0 | **FeaturesEdx**. Read-write. Provides back-door control over the features reported in CPUID function 1, EDX (see CPUID Fn0000_0001_EDX). |

### MSRC001_1005 Extended CPUID Features Register (ExtFeatures)

See MSRC001_1004.

| Bits | Description |
|---|---|
| 63:32 | **ExtFeaturesEcx**. Read-write. Provides back-door control over the features reported in CPUID function 8000_0001, ECX (see CPUID Fn8000_0001_ECX). |
| 31:0 | **ExtFeaturesEdx**. Read-write. Provides back-door control over the features reported in CPUID function 8000_0001, EDX (see CPUID Fn8000_0001_EDX). |

### MSRC001_1020 Load-Store Configuration Register (LS_CFG)

Reset: 0002_0010_0000_1000h.

| Bits | Description |
|---|---|
| 63:29 | Reserved. |
| 28 | **DIS_SS**. Read-write. 1=Disable streaming store functionality. See 2.3.3 [Cache Initialization For General Storage During Boot]. |
| 27:0 | Reserved. |

### MSRC001_1021 Instruction Cache Configuration Register (IC_CFG)

Reset: 0000_0000_0000_0000h.

| Bits | Description |
|---|---|
| 63:10 | Reserved. |
| 9 | **DIS_SPEC_TLB_RLD**. Read-write. BIOS: See 2.3.3 [Cache Initialization For General Storage During Boot]. 1=Disable speculative TLB reloads. |
| 8:0 | Reserved. |

### MSRC001_1022 Data Cache Configuration Register (DC_CFG)

| Bits | Description |
|---|---|
| 63:14 | Reserved. |
| 13 | **DIS_HW_PF**. Read-write. Reset: 0. BIOS: See 2.3.3 [Cache Initialization For General Storage During Boot]. 1=Disable hardware prefetches. |
| 12:9 | Reserved. |
| 8 | **DIS_CLR_WBTOL2_SMC_HIT**. Read-write. Reset: 0.1=Disable self modifying code checking logic. See 2.3.3 [Cache Initialization For General Storage During Boot]. |
| 7:5 | Reserved. |
| 4 | **DIS_SPEC_TLB_RLD**. Read-write. Reset: 0. 1=Disable speculative TLB reloads. BIOS: See 2.3.3 [Cache Initialization For General Storage During Boot]. |
| 3:0 | Reserved. |

### MSRC001_1029 Decode Configuration Register (DE_CFG)

Reset: 0000_0000_0000_0000h.

| Bits | Description |
|---|---|
| 63:24 | Reserved. |
| 23 | **ClflushSerialize**. Read-write. 1=CLFLUSH is a serializing instruction. 0=CLFLUSH is not a serializing instruction. |
| 22:0 | Reserved. |

### MSRC001_102A Bus Unit Configuration 2 Register (BU_CFG2)

Reset: 0000_1840_0100_8800h.

| Bits | Description |
|---|---|
| 63:59 | Reserved. |
| 58:57 | **L2HystCnt: L2 clock gating hysteresis**. Read-write. BIOS: 11b. Specifies the maximum number of clocks the L2 is gated before a periodic clock must be sent to prevent SOI hysteresis.<br>Bits — Definition<br>00b — No periodic clocks.<br>01b — 1 periodic clock every 8 core clocks.<br>10b — 1 periodic clock every 16 core clocks.<br>11b — 1 periodic clock every 32 core clocks |
| 56 | **L2ClkGatingEn: enable L2 clock gating**. Read-write. BIOS: 1. 1=L2 clock gating enabled. |
| 55:51 | Reserved. |
| 50 | **RdMmExtCfgDwDis: read MMIO extended config double word disable**. Read-write. BIOS: 1. 1=MMIO reads to extended config space do not need to be double word aligned and may be up to quadword sized. 0=MMIO reads to extended config space must be doubleword aligned and doubleword size or smaller. |
| 49:36 | Reserved. |
| 35 | **IcDisSpecTlbWr**. Read-write. 1=Speculative writes to the ITLB by the TLB reloader are disabled. |
| 34:0 | Reserved. |

### MSRC001_1030 IBS Fetch Control Register (IbsFetchCtl)

Reset: 0000_0000_0000_0000h. The IBS fetch sampling engine selects an instruction fetch to profile when the engine's periodic fetch counter reaches IbsFetchMaxCnt. The periodic fetch counter is an internal 20 bit counter that increments after every fetch cycle that completes when IbsFetchEn=1 and IbsFetchVal=0. When the selected instruction fetch completes or is aborted, the status of the fetch is written to the IBS fetch registers (this register, MSRC001_1031 and MSRC001_1032) and an interrupt is generated. The interrupt service routine associated with this interrupt is responsible for saving the performance information stored in IBS fetch registers. See 2.6.2 [Instruction Based Sampling (IBS)].

| Bits | Description |
|---|---|
| 63:58 | Reserved. |

| | |
|---|---|
| 57 | **IbsRandEn: random instruction fetch tagging enable**. Read-write. 1=Bits 3:0 of the fetch counter are randomized when the IbsFetchCtl register is written. 0=Bits 3:0 of the fetch counter are set to 0h when IbsFetchEn is set to start the fetch counter. |
| 56 | **IbsL2TlbMiss: instruction cache L2TLB miss**. Read-write. 1=The instruction fetch missed in the L2 TLB. |
| 55 | **IbsL1TlbMiss: instruction cache L1TLB miss**. Read-write. 1=The instruction fetch missed in the L1 TLB. |
| 54:53 | **IbsL1TlbPgSz: instruction cache L1TLB page size**. Read-write. This field indicates the page size of the translation in the L1 TLB. This field is only valid if IbsPhyAddrValid=1.<br>Bits — Definition<br>00b — 4 Kbyte<br>01b — 2 Mbyte<br>10b — 1 Gbyte<br>11b — Reserved |
| 52 | **IbsPhyAddrValid: instruction fetch physical address valid**. Read-write. 1=The physical address in MSRC001_1032 and the IbsL1TlbPgSz field are valid for the instruction fetch. |
| 51 | **IbsIcMiss: instruction cache miss**. Read-write. 1=The instruction fetch missed in the instruction cache. |
| 50 | **IbsFetchComp: instruction fetch complete**. Read-write. 1=The instruction fetch completed and the data is available for use by the instruction decoder. |
| 49 | **IbsFetchVal: instruction fetch valid**. Read-write; set-by-hardware. 1=New instruction fetch data available. When this bit is set, the periodic fetch counter stops counting until software clears the bit and an interrupt is generated as specified by MSRC001_103A. |
| 48 | **IbsFetchEn: instruction fetch enable**. Read-write. 1=Instruction fetch sampling is enabled. |
| 47:32 | **IbsFetchLat: instruction fetch latency**. Read-write. This field indicates the number of clock cycles from when the instruction fetch was initiated to when the data was delivered to the core. If the instruction fetch is abandoned before the fetch completes, this field returns the number of clock cycles from when the instruction fetch was initiated to when the fetch was abandoned. |
| 31:16 | **IbsFetchCnt**. Read-write; updated-by-hardware. This field returns the current value of bits 19:4 of the periodic fetch counter. |
| 15:0 | **IbsFetchMaxCnt**. Read-write. This field specifies maximum count value of the periodic fetch counter. Programming this field to 0000h and setting IbsFetchEn results in undefined behavior. Bits 19:4 of the maximum count are programmed in the field. Bits 3:0 of the maximum count are always 0000b. |

### MSRC001_1031 IBS Fetch Linear Address Register (IbsFetchLinAd)

Reset: 0000_0000_0000_0000h.

| Bits | Description |
|---|---|
| 63:0 | **IbsFetchLinAd: instruction fetch linear address**. Read-write. This field provides the linear address in canonical form for the tagged instruction fetch. |

### MSRC001_1032 IBS Fetch Physical Address Register (IbsFetchPhysAd)

Reset: 0000_0000_0000_0000h.

| Bits | Description |
|---|---|
| 63:0 | **IbsFetchPhysAd: instruction fetch physical address**. Read-write. This provides the physical address in for the tagged instruction fetch. The lower 12 bits are not modified by address translation, so they are always the same as the linear address. This field contains valid data only if MSRC001_1030[IbsPhyAddrValid] is asserted. |

### MSRC001_1033 IBS Execution Control Register (IbsOpCtl)

Reset: 0000_0000_0000_0000h. The IBS execution sampling engine tags a micro-op that is issued in the next cycle to profile when the engine's periodic op counter reaches IbsOpMaxCnt. The periodic op counter is an internal 20 bit counter that increments every cycle when IbsOpEn=1 and IbsOpVal=0 and rolls over when the counter reaches IbsOpMaxCnt. When the periodic op counter rolls over bits 6:0 of the counter are randomized by hardware. When the micro-op is retired, the status of the operation is written to the IBS execution registers (this register, MSRC001_1034, MSRC001_1035, MSRC001_1036, MSRC001_1037, MSRC001_1038 and MSRC001_1039) and an interrupt is generated as specified by MSRC001_103A. The interrupt service routine associated with this interrupt is responsible for saving the performance information stored in IBS execution registers. See 2.6.2 [Instruction Based Sampling (IBS)].

| Bits | Description |
|---|---|
| 63:59 | Reserved. |
| 58:52 | **IbsOpCurCntExt: periodic op counter current count extension**. Read-write. This field returns the current value of bits 26:20 of the periodic op counter. |
| 51:32 | **IbsOpCurCnt: periodic op counter current count**. Read-write. This field returns the current value of the periodic op counter. |
| 31:27 | Reserved. |
| 26:20 | **IbsOpMaxCntExt: periodic op counter maximum count extension**. Read-write. This field specifies maximum count value of the periodic op counter. Bits 26:20 of the maximum count are programmed in the field. |
| 19 | **IbsOpCntCtl: periodic op counter count control**. Read-write. |
| 18 | **IbsOpVal: micro-op sample valid**. Read-write; set-by-hardware.1=New instruction execution data available. When this bit is set, the periodic op counter stops counting until software clears the bit and an interrupt is generated as specified by MSRC001_103A. |
| 17 | **IbsOpEn: micro-op sampling enable**. Read-write. 1=Instruction execution sampling enabled. |
| 16 | Reserved. |
| 15:0 | **IbsOpMaxCnt: periodic op counter maximum count**. Read-write. This field specifies maximum count value of the periodic op counter. Bits 19:4 of the maximum count are programmed in the field. Bits 3:0 of the maximum count are always 0000. |

### MSRC001_1034 IBS Op Logical Address Register (IbsOpRip)

Reset: 0000_0000_0000_0000h.

| Bits | Description |
|---|---|
| 63:0 | **IbsOpRip: micro-op logical address**. Read-write. Logical address in canonical form for the instruction that contains the tagged micro-op. |

### MSRC001_1035 IBS Op Data Register (IbsOpData)

Reset: 0000_0000_0000_0000h.

| Bits | Description |
|---|---|
| 63:39 | Reserved. |
| 38 | **IbsRipInvalid: RIP invalid**. Read-write. 1=MSRC001_1034[IbsOpRip] is not valid for the tagged micro-op. |
| 37 | **IbsOpBrnRet: branch micro-op retired**. Read-write. 1=Tagged operation was a branch micro-op that retired. |
| 36 | **IbsOpBrnMisp: mispredicted branch micro-op**. Read-write. 1=Tagged operation was a branch micro-op that was mispredicted. |
| 35 | **IbsOpBrnTaken: taken branch micro-op**. Read-write. 1=Tagged operation was a branch micro-op that was taken. |
| 34 | **IbsOpReturn: return micro-op**. Read-write. 1=Tagged operation was return micro-op. |
| 33 | **IbsOpMispReturn: mispredicted return micro-op**. Read-write. 1=Tagged operation was a mispredicted return micro-op. |
| 32 | **IbsOpBrnResync: resync micro-op**. Read-write. 1=Tagged operation was resync micro-op. |
| 31:16 | **IbsTagToRetCtr: micro-op tag to retire count**. Read-write. This field returns the number of cycles from when the micro-op was tagged to when the micro-op was retired. This field is equal to IbsCompToRetCtr when the tagged micro-op is a NOP. |
| 15:0 | **IbsCompToRetCtr: micro-op completion to retire count**. Read-write. This field returns the number of cycles from when the micro-op was completed to when the micro-op was retired. |

### MSRC001_1036 IBS Op Data 2 Register (IbsOpData2)

Northbridge data is only valid for load operations that miss both the L1 data cache and the L2 cache. If a load operation crosses a cache line boundary, the data returned in this register is the data for the access to the lower cache line.

| Bits | Description |
|---|---|
| 63:32 | Reserved. |
| 31:6 | Reserved. |
| 5 | **NbIbsReqCacheHitSt**: **IBS cache hit state**. Read-only. Value: 0. Valid when the data source type is Cache(2h). 0=M State. 1=O State. |

| | |
|---|---|
| 4 | **NbIbsReqDstNode**: **IBS request destination node**. Read-only. Value: 0. 0=The request is serviced by the NB in the same node as the core. 1=The request is serviced by the NB in a different node than the core. Valid when NbIbsReqSrc is non-zero. |
| 3 | Reserved. |
| 2:0 | **NbIbsReqSrc: northbridge IBS request data source**. Read-write; updated-by-hardware. Reset: 0.<br>Bits — Definition<br>0h — No valid status<br>1h — Reserved<br>2h — Cache: data returned from a CPU cache<br>3h — DRAM: data returned from DRAM<br>4h — Reserved<br>5h — Reserved<br>6h — Reserved<br>7h — Other: data returned from MMIO/Config/PCI/APIC |

### MSRC001_1037 IBS Op Data 3 Register (IbsOpData3)

Reset: 0000_0000_0000_0000h. If a load or store operation crosses a 128-bit boundary, the data returned in this register is the data for the access to the data below the 128-bit boundary.

| Bits | Description |
|---|---|
| 63:48 | Reserved. |
| 47:32 | **IbsDcMissLat: data cache miss latency**. Read-write. This field indicates the number of clock cycles from when a miss is detected in the data cache to when the data was delivered to the core. The value returned by this counter is not valid for data cache writes. |
| 31:20 | Reserved. |
| 19 | **IbsDcL2tlbHit1G: data cache L2TLB hit in 1G page**. Read-write. 1=The physical address for the tagged load or store operation was present in a 1G page table entry in the data cache L2TLB. |
| 18 | **IbsDcPhyAddrValid: data cache physical address valid**. Read-write. 1=The physical address in MSRC001_1039 is valid for the load or store operation. |
| 17 | **IbsDcLinAddrValid: data cache linear address valid**. Read-write. 1=The linear address in MSRC001_1038 is valid for the load or store operation. |
| 16 | **IbsDcMabHit: MAB hit**. Read-write. 1=The tagged load or store operation hit on an already allocated MAB. IBS data in MSRC001_1036 is not valid if this bit is set. |
| 15 | **IbsDcLockedOp: locked operation**. Read-write. 1=Tagged load or store operation is a locked operation. |
| 14 | **IbsDcUcMemAcc: UC memory access**. Read-write. 1=Tagged load or store operation accessed uncacheable memory. |
| 13 | **IbsDcWcMemAcc: WC memory access**. Read-write. 1=Tagged load or store operation accessed write combining memory. |
| 12 | **IbsDcStToLdCan: data forwarding from store to load operation cancelled**. Read-write. 1=Data forwarding from a store operation to the tagged load was cancelled. |
| 11 | **IbsDcStToLdFwd: data forwarded from store to load operation**. Read-write. 1=Data for tagged load operation was forwarded from a store operation. If this bit is set and IbsDcStToLdCan=1, then the data for the load operation forwarded from a store operation but the data was not forwarded immediately. |

| | |
|---|---|
| 10 | **IbsDcStBnkCon: bank conflict on store operation**. Read-write. 1=A bank conflict with a store operation occurred in the data cache on the tagged load or store operation. |
| 9 | **IbsDcLdBnkCon: bank conflict on load operation**. Read-write. 1=A bank conflict with a load operation occurred in the data cache on the tagged load or store operation. |
| 8 | **IbsDcMisAcc: misaligned access**. Read-write. 1=The tagged load or store operation crosses a 128 bit address boundary. |
| 7 | **IbsDcMiss: data cache miss**. Read-write. 1=The cache line used by the tagged load or store was not present in the data cache. |
| 6 | **IbsDcL2tlbHit2M: data cache L2TLB hit in 2M page**. Read-write. 1=The physical address for the tagged load or store operation was present in a 2M page table entry in the data cache L2TLB. |
| 5 | **IbsDcL1tlbHit1G: data cache L1TLB hit in 1G page**. Read-write. 1=The physical address for the tagged load or store operation was present in a 1G page table entry in the data cache L1TLB. |
| 4 | **IbsDcL1tlbHit2M: data cache L1TLB hit in 2M page**. Read-write. 1=The physical address for the tagged load or store operation was present in a 2M page table entry in the data cache L1TLB. |
| 3 | **IbsDcL2tlbMiss: data cache L2TLB miss**. Read-write. 1=The physical address for the tagged load or store operation was not present in the data cache L2TLB. |
| 2 | **IbsDcL1tlbMiss: data cache L1TLB miss**. Read-write. 1=The physical address for the tagged load or store operation was not present in the data cache L1TLB. |
| 1 | **IbsStOp: store op**. Read-write. 1=Tagged operation is a store operation |
| 0 | **IbsLdOp: load op**. Read-write. 1=Tagged operation is a load operation |

### MSRC001_1038 IBS DC Linear Address Register (IbsDcLinAd)

Reset: 0000_0000_0000_0000h.

| Bits | Description |
|---|---|
| 63:0 | **IbsDcLinAd**. Read-write. This field provides the linear address in canonical form for the tagged load or store operation. This field contains valid data only if MSRC001_1037[IbsDcLinAddrValid] is asserted. |

### MSRC001_1039 IBS DC Physical Address Register (IbsDcPhysAd)

Reset: 0000_0000_0000_0000h.

| Bits | Description |
|---|---|
| 63:0 | **IbsDcPhysAd: load or store physical address**. Read-write. Provides the physical address for the tagged load or store operation. The lower 12 bits are not modified by address translation, so they are always the same as the linear address. This field contains valid data only if MSRC001_1037[IbsDcPhyAddrValid] is asserted. |

### MSRC001_103A IBS Control Register

Reset: 0000_0000_0000_0000h. Read-only. This register returns the value of D18F3x1CC.

| Bits | Description |
|---|---|
| 63:9 | Reserved. |

| | |
|---|---|
| 8 | **LvtOffsetVal: local vector table offset valid**. Read-only. 1=The offset in LvtOffset is valid. |
| 7:4 | Reserved. |
| 3:0 | **LvtOffset: local vector table offset**. Read-only. This specifies the address of the IBS LVT entry in the APIC registers as follows: LVT address = (LvtOff shifted left 4 bits) + 500h (see APIC[530:500]). |

**MSRC001_103B IBS Branch Target Address**

Reset: 0000_0000_0000_0000h. Read-only. This register returns the value of D18F3x1CC.

| Bits | Description |
|---|---|
| 63:0 | **IbsBrTarget**. Read-write. This provides the logical address in canonical form for the branch target. This field contains a valid branch target address when the tagged operation is a branch and the field is non-zero. |

## 3.24 Performance Counter Events

This section provides the performance counter events that may be selected through [The Performance Event Select Register (PERF_CTL[3:0])] MSRC001_00[03:00][EventSelect and UnitMask]. See MSRC001_00[03:00] and [The Performance Event Counter Registers (PERF_CTR[3:0])] MSRC001_00[07:04].

### 3.24.1 Floating Point Events

See the following events for additional floating point information:

- PMCx0CB [Retired MMX™/FP Instructions].
- PMCx0DB [FPU Exceptions].
- PMCx1C0 [Retired x87 Floating Point Operations].

**PMCx000 Dispatched FPU Operations**

The number of operations (uops) dispatched to the FPU execution pipelines. This event reflects how busy the FPU pipelines are. This includes all operations done by x87, MMX™ and SSE instructions, including moves. Each increment represents a one-cycle dispatch event; packed 128-bit SSE operations count as two ops in 64-bit FPU implementations; scalar operations count as one. This event is a speculative event. (See PMCx0CB). Since this event includes non-numeric operations it is not suitable for measuring MFLOPs.

| UnitMask | Description |
|---|---|
| 7:6 | Reserved. |
| 5 | Store pipe load ops and SSE move ops |
| 4 | Multiply pipe load ops and SSE move ops |
| 3 | Add pipe load ops and SSE move ops |
| 2 | Store pipe ops excluding load ops and SSE move ops |
| 1 | Multiply pipe ops excluding load ops and SSE move ops |
| 0 | Add pipe ops excluding load ops and SSE move ops |

**PMCx001 Cycles in which the FPU is Empty**

The number of cycles in which the FPU is empty. Invert this (MSRC001_00[03:00][Invert]=1) to count cycles in which at least one FPU operation is present in the FPU.

### PMCx002 Dispatched Fast Flag FPU Operations

The number of FPU operations that use the fast flag interface (e.g. FCOMI, COMISS, COMISD, UCOMISS, UCOMISD, MOVD, CVTSD2SI). This event is a speculative event.

### PMCx003 Retired SSE Operations

The number of SSE operations retired. This counter can count either FLOPS (UnitMask bit 6 = 1) or uops (UnitMask bit 6 = 0).

| UnitMask | Description |
|---|---|
| 7 | Reserved. |
| 6 | Op type: 0=uops. 1=FLOPS |
| 5 | Double precision divide/square root ops |
| 4 | Double precision multiply ops |
| 3 | Double precision add/subtract ops |
| 2 | Single precision divide/square root ops |
| 1 | Single precision multiply ops |
| 0 | Single precision add/subtract ops |

### PMCx004 Retired Move Ops

The number of move uops retired. Merging low quadword move ops copy the lower 64 bits of a source register to the upper 64 bits of a destination register. The lower 64 bits of the destination register remain unchanged. Merging high quadword move ops copy the upper 64 bits of a source register to the lower 64 bits of a destination register. The upper 64 bits of the destination register remain unchanged.

| UnitMask | Description |
|---|---|
| 7:4 | Reserved. |
| 3 | All other move uops |
| 2 | All other merging move uops |
| 1 | Merging high quadword move uops |
| 0 | Merging low quadword move uops |

### PMCx005 Retired Serializing Ops

The number of serializing uops retired. A bottom-executing uop is not issued until it is the oldest non-retired uop in the FPU. Bottom-executing ops are most commonly seen with FSTSW and STMXCSR instructions. A bottom-serializing uop does not issue until it is the oldest non-issued uop in the FP scheduler. Bottom-serializing uops block all subsequent uops from being issued until the uop is issued. Bottom-serializing ops are most commonly seen with FLCDW and LDMXCSR instructions.

| UnitMask | Description |
|---|---|
| 7:4 | Reserved. |

| | |
|---|---|
| 3 | x87 bottom-serializing uops retired |
| 2 | x87 bottom-executing uops retired |
| 1 | SSE bottom-serializing uops retired |
| 0 | SSE bottom-executing uops retired |

**PMCx006 Number of Cycles that a Serializing uop is in the FP Scheduler**

See PMCx005 for a description of bottom-executing and bottom-serializing uop.

| UnitMask | Description |
|---|---|
| 7:2 | Reserved. |
| 1 | Number of cycles a bottom-serializing uop is in the FP scheduler |
| 0 | Number of cycles a bottom-execute uop is in the FP scheduler |

### 3.24.2 Load/Store and TLB Events

See the following events for additional Load/Store and TLB information:

- PMCx065 [Memory Requests by Type].

**PMCx020 Segment Register Loads**

The number of segment register loads performed.

| UnitMask | Description |
|---|---|
| 7 | Reserved. |
| 6 | HS |
| 5 | GS |
| 4 | FS |
| 3 | DS |
| 2 | SS |
| 1 | CS |
| 0 | ES |

**PMCx021 Pipeline Restart Due to Self-Modifying Code**

The number of pipeline restarts caused by self-modifying code (a store that hits any instruction that has been fetched for execution beyond the instruction doing the store).

**PMCx022 Pipeline Restart Due to Probe Hit**

The number of pipeline restarts caused by an invalidating probe hitting on a speculative out-of-order load.

**PMCx023 LS Buffer 2 Full**

The number of cycles that the LS2 buffer is full. This buffer holds stores waiting to retire as well as requests that missed the data cache and are waiting on a refill. This condition stalls further data cache accesses, although such stalls may be overlapped by independent instruction execution.

**PMCx024 Locked Operations**

This event covers locked operations performed and their execution time. The execution time represented by the cycle counts is typically overlapped to a large extent with other instructions. The non-speculative cycles event is suitable for event-based profiling of lock operations that tend to miss in the cache.

| UnitMask | Description |
|---|---|
| 7:4 | Reserved. |
| 3 | The number of cycles waiting for a cache hit (cache miss penalty). |
| 2 | The number of cycles spent in non-speculative phase (including cache miss penalty) |
| 1 | The number of cycles spent in speculative phase |
| 0 | The number of locked instructions executed |

**PMCx026 Retired CLFLUSH Instructions**

The number of CLFLUSH instructions retired.

**PMCx027 Retired CPUID Instructions**

The number of CPUID instructions retired.

**PMCx02A Cancelled Store to Load Forward Operations**

Counts the number store to load forward operations that are cancelled.

| UnitMask | Description |
|---|---|
| 7:3 | Reserved. |
| 2 | Misaligned. |
| 1 | Store is smaller than load. |
| 0 | Address mismatches (starting byte not the same). |

**PMCx02B SMIs Received**

Counts the number of SMIs received by the processor.

### 3.24.3 Data Cache Events

**PMCx040 Data Cache Accesses**

The number of accesses to the data cache for load and store references. This may include certain microcode scratchpad accesses, although these are generally rare. Each increment represents an eight-byte access, although the instruction may only be accessing a portion of that. This event is a speculative event.

**PMCx041 Data Cache Misses**

The number of data cache references which missed in the data cache. This event is a speculative event.

**PMCx042 Data Cache Refills from L2 or Northbridge**

The number of data cache refills satisfied from the L2 cache (and/or the northbridge), per the UnitMask. Unit-Mask bits 4:1 allow a breakdown of refills from the L2 by coherency state. UnitMask bit 0 reflects refills

which missed in the L2, and provides the same measure as the combined sub-events of PMCx043. Each increment reflects a 64-byte transfer. This event is a speculative event.

| UnitMask | Description |
|---|---|
| 7:5 | Reserved. |
| 4 | Modified-state line from L2 |
| 3 | Owned-state line from L2 |
| 2 | Exclusive-state line from L2 |
| 1 | Shared-state line from L2 |
| 0 | Refill from the northbridge |

**PMCx043 Data Cache Refills from the northbridge**

The number of L1 cache refills satisfied from the northbridge (DRAM or another processor's cache), as opposed to the L2. The UnitMask selects lines in one or more specific coherency states. Each increment reflects a 64-byte transfer. This event is a speculative event.

| UnitMask | Description |
|---|---|
| 7:5 | Reserved. |
| 4 | Modified |
| 3 | Owned |
| 2 | Exclusive |
| 1 | Shared |
| 0 | Invalid |

**PMCx044 Data Cache Lines Evicted**

The number of L1 data cache lines written to the L2 cache or system memory, having been displaced by L1 refills. The UnitMask may be used to count only victims in specific coherency states. Each increment represents a 64-byte transfer. This event is a speculative event.

In most cases, L1 victims are moved to the L2 cache, displacing an older cache line there. Lines brought into the data cache by PrefetchNTA instructions, however, are evicted directly to system memory (if dirty) or invalidated (if clean). The Invalid case (UnitMask[0]) reflects the replacement of lines that would have been invalidated by probes for write operations from another processor or DMA activity. UnitMask[2,1] count all evictions regardless of cache line state. When either UnitMask[2 or1] is enabled all other UnitMasks should be disabled.

| UnitMask | Description |
|---|---|
| 7 | Reserved. |
| 6 | Cache line evicted was not brought into the cache with by a PrefetchNTA instruction. |
| 5 | Cache line evicted was brought into the cache with by a PrefetchNTA instruction. |
| 4 | Modified |
| 3 | Owned |
| 2 | Exclusive |

| | |
|---|---|
| 1 | Shared |
| 0 | Invalid |

### PMCx045 L1 DTLB Miss and L2 DTLB Hit

The number of data cache accesses that miss in the L1 DTLB and hit in the L2 DTLB. This event is a speculative event.

| UnitMask | Description |
|---|---|
| 7:3 | Reserved. |
| 2 | L2 1G TLB hit |
| 1 | L2 2M TLB hit |
| 0 | L2 4K TLB hit |

### PMCx046 L1 DTLB and L2 DTLB Miss

The number of data cache accesses that miss in both the L1 and L2 DTLBs. This event is a speculative event.

| UnitMask | Description |
|---|---|
| 7:3 | Reserved. |
| 2 | 1G TLB reload |
| 1 | 2M TLB reload |
| 0 | 4K TLB reload |

### PMCx047 Misaligned Accesses

The number of data cache accesses that are misaligned. These are accesses which cross a sixteen-byte boundary. They incur an extra cache access (reflected in PMCx040), and an extra cycle of latency on reads. This event is a speculative event.

### PMCx048 Microarchitectural Late Cancel of an Access

### PMCx049 Microarchitectural Early Cancel of an Access

### PMCx04B Prefetch Instructions Dispatched

The number of prefetch instructions dispatched by the decoder. Such instructions may or may not cause a cache line transfer. All Dcache and L2 accesses, hits and misses by prefetch instructions, except for prefetch instructions that collide with an outstanding hardware prefetch, are included in these events. This event is a speculative event.

| UnitMask | Description |
|---|---|
| 7:3 | Reserved. |
| 2 | NTA (PrefetchNTA) |

| | |
|---|---|
| 1 | Store (PrefetchW) |
| 0 | Load (Prefetch, PrefetchT0/T1/T2) |

#### PMCx04C DCACHE Misses by Locked Instructions

The number of data cache misses incurred by locked instructions. (The total number of locked instructions may be obtained from PMCx024.)

Such misses may be satisfied from the L2 or system memory, but there is no provision for distinguishing between the two. When used for event-based profiling, this event tends to occur very close to the offending instructions. This event is also included in the basic Dcache miss event (PMCx041).

| UnitMask | Description |
|---|---|
| 7:2 | Reserved. |
| 1 | Data cache misses by locked instructions |
| 0 | Reserved. |

#### PMCx04D L1 DTLB Hit

The number of data cache accesses that hit in the L1 DTLB. This event is a speculative event.

| UnitMask | Description |
|---|---|
| 7:3 | Reserved. |
| 2 | L1 1G TLB hit |
| 1 | L1 2M TLB hit |
| 0 | L1 4K TLB hit |

#### PMCx052 Ineffective Software Prefetches

The number of software prefetches that did not fetch data outside of the processor core.

| UnitMask | Description |
|---|---|
| 7:4 | Reserved. |
| 3 | Software prefetch hit in L2. |
| 2:1 | Reserved. |
| 0 | Software prefetch hit in the L1. |

#### PMCx054 Global TLB Flushes

This event counts TLB flushes that flush TLB entries that have the global bit set.

### 3.24.4 L2 Cache and System Interface Events

#### PMCx065 Memory Requests by Type

These events reflect accesses to uncacheable (UC) or write-combining (WC) memory regions (as defined by MTRR or PAT settings) and Streaming Store activity to WB memory. Both the WC and Streaming Store events reflect Write Combining buffer flushes, not individual store instructions. WC buffer flushes which typically

consist of one 64-byte write to the system for each flush (assuming software typically fills a buffer before it gets flushed). A partially-filled buffer requires two or more smaller writes to the system. The WC event reflects flushes of WC buffers that are filled by stores to WC memory or streaming stores to WB memory. The Streaming Store event reflects only flushes due to streaming stores (which are typically only to WB memory). The difference between counts of these two events reflects the true amount of write events to WC memory.

| UnitMask | Description |
|---|---|
| 7 | Streaming store (SS) requests |
| 6:3 | Reserved. |
| 2 | Requests to cache-disabled (CD) memory |
| 1 | Requests to write-combining (WC) memory or WC buffer flushes to WB memory |
| 0 | Requests to non-cacheable (UC) memory |

### PMCx067 Data Prefetcher

These events reflect requests made by the data prefetcher. UnitMask[1] counts total prefetch requests, while bit 0 counts requests where the target block is found in the L2 or data cache. The difference between the two represents actual data read (in units of 64-byte cache lines) from the system by the prefetcher. This is also included in the count of PMCx07F, UnitMask[0] (combined with other L2 fill events).

| UnitMask | Description |
|---|---|
| 7:2 | Reserved. |
| 1 | Prefetch attempts |
| 0 | Cancelled prefetches |

### PMCx06C Northbridge Read Responses by Coherency State

The number of responses from the Northbridge for cache refill requests. The UnitMask may be used to select specific cache coherency states. Each increment represents one 64-byte cache line transferred from the Northbridge (DRAM or another cache) to the data cache, instruction cache or L2 cache (for data prefetcher and TLB table walks). Modified-state responses may be for Dcache store miss refills, PrefetchW software prefetches, hardware prefetches for a store-miss stream, or Change-to-Dirty requests that get a dirty (Owned) probe hit in another cache. Exclusive responses may be for any Icache refill, Dcache load miss refill, other software prefetches, hardware prefetches for a load-miss stream, or TLB table walks that miss in the L2 cache; Shared responses may be for any of those that hit a clean line in another cache.

| UnitMask | Description |
|---|---|
| 7:5 | Reserved. |
| 4 | Data Error |
| 3 | Owned |
| 2 | Shared |
| 1 | Modified |
| 0 | Exclusive |

### PMCx06D Octwords Written to System

The number of octword (16-byte) data transfers from the processor to the system. These may be part of a 64-byte cache line writeback or a 64-byte dirty probe hit response, each of which would cause four increments; or a partial or complete Write Combining buffer flush (Sized Write), which could cause from one to four increments.

| UnitMask | Description |
|---|---|
| 7:1 | Reserved. |
| 0 | Octword write transfer |

### PMCx076 CPU Clocks not Halted

The number of clocks that the CPU is not in a halted state (due to STPCLK or a HALT instruction). This event allows system idle time to be automatically factored out from IPC (or CPI) measurements, providing the OS halts the CPU when going idle. If the OS goes into an idle loop rather than halting, such calculations are influenced by the IPC of the idle loop.

### PMCx07D Requests to L2 Cache

The number of requests to the L2 cache for Icache or Dcache fills, or page table lookups for the TLB. These events reflect only read requests to the L2. These include some amount of retries associated with address or resource conflicts. Such retries tend to occur more as the L2 gets busier, and in certain extreme cases (such as large block moves that overflow the L2) these extra requests can dominate the event count.

These extra requests are not a direct indication of performance impact - they simply reflect opportunistic accesses that don't complete. But because of this, they are not a good indication of actual cache line movement. The Icache and Dcache miss and refill events (81h, 82h, 83h, 41h, 42h, 43h) provide a more accurate indication of this, and are the preferred way to measure such traffic.

| UnitMask | Description |
|---|---|
| 7:6 | Reserved. |
| 5 | Hardware prefetch from DC |
| 4 | Cancelled request |
| 3 | Tag snoop request |
| 2 | TLB fill (page table walks) |
| 1 | DC fill |
| 0 | IC fill |

### PMCx07E L2 Cache Misses

The number of requests that miss in the L2 cache. This may include some amount of speculative activity, as well as some amount of retried requests as described in PMCx07D. The IC-fill-miss and DC-fill-miss events tend to mirror the Icache and Dcache refill-from-system events (83h and PMCx043, respectively), and tend to include more speculative activity than those events.

| UnitMask | Description |
|---|---|
| 7:4 | Reserved. |

| | |
|---|---|
| 3 | Hardware prefetch from DC |
| 2 | TLB page table walk |
| 1 | DC fill (includes possible replays, whereas PMCx041 does not) |
| 0 | IC fill |

**PMCx07F L2 Fill/Writeback**

The number of lines written into the L2 cache due to victim writebacks from the Icache or Dcache, TLB page table walks and the hardware data prefetcher (UnitMask[0]); or writebacks of dirty lines from the L2 to the system (UnitMask[1]). Each increment represents a 64-byte cache line transfer.

Victim writebacks from the Dcache may be measured separately using PMCx044. However this is not quite the same as the Dcache component of this event, the main difference being PrefetchNTA lines. When these are evicted from the Dcache due to replacement, they are written out to system memory (if dirty) or simply invalidated (if clean), rather than being moved to the L2 cache.

| UnitMask | Description |
|---|---|
| 7:2 | Reserved. |
| 1 | L2 Writebacks to system. |
| 0 | L2 fills (victims from L1 caches, TLB page table walks and data prefetches) |

**PMCx165 Page Size Mismatches**

Counts the number of large pages that are installed into the TLB as a smaller page size.

| UnitMask | Description |
|---|---|
| 7:3 | Reserved. |
| 2 | Host page size is larger than the guest page size. |
| 1 | MTRR mismatch. |
| 0 | Guest page size is larger than the host page size. |

### 3.24.5 Instruction Cache Events

All instruction cache events are speculative events unless specified otherwise.

**PMCx080 Instruction Cache Fetches**

The number of instruction cache accesses by the instruction fetcher. Each access is an aligned 32 byte read, from which a varying number of instructions may be decoded.

**PMCx081 Instruction Cache Misses**

The number of instruction fetches and prefetch requests that miss in the instruction cache. This is typically equal to or very close to the sum of events 82h and 83h. Each miss results in a 64-byte cache line refill.

**PMCx082 Instruction Cache Refills from L2**

The number of instruction cache refills satisfied from the L2 cache. Each increment represents one 64-byte cache line transfer.

### PMCx083 Instruction Cache Refills from System

The number of instruction cache refills from system memory (or another cache). Each increment represents one 64-byte cache line transfer.

### PMCx084 L1 ITLB Miss, L2 ITLB Hit

The number of instruction fetches that miss in the L1 ITLB but hit in the L2 ITLB.

### PMCx085 L1 ITLB Miss, L2 ITLB Miss

The number of instruction fetches that miss in both the L1 and L2 ITLBs.

| UnitMask | Description |
|---|---|
| 7:2 | Reserved. |
| 1 | Instruction fetches to a 2M page. |
| 0 | Instruction fetches to a 4K page. |

### PMCx086 Pipeline Restart Due to Instruction Stream Probe

The number of pipeline restarts caused by invalidating probes that hit on the instruction stream currently being executed. This would happen if the active instruction stream was being modified by another processor in an MP system - typically a highly unlikely event.

### PMCx087 Instruction Fetch Stall

The number of cycles the instruction fetcher is stalled. This may be for a variety of reasons such as branch predictor updates, unconditional branch bubbles, far jumps and cache misses, among others. May be overlapped by instruction dispatch stalls or instruction execution, such that these stalls don't necessarily impact performance.

### PMCx088 Return Stack Hits

The number of near return instructions (RET or RET Iw) that get their return address from the return address stack (i.e. where the stack has not gone empty). This may include cases where the address is incorrect (return mispredicts). This may also include speculatively executed false-path returns. Return mispredicts are typically caused by the return address stack underflowing, however they may also be caused by an imbalance in calls vs. returns, such as doing a call but then popping the return address off the stack.

This event cannot be reliably compared with events C9h and CAh (such as to calculate percentage of return mispredicts due to an empty return address stack), since it may include speculatively executed false-path returns that are not included in those retire-time events.

### PMCx089 Return Stack Overflows

The number of (near) call instructions that cause the return address stack to overflow. When this happens, the oldest entry is discarded. This count may include speculatively executed calls.

### PMCx08B Instruction Cache Victims

The number of cachelines evicted from the instruction cache to the L2.

### PMCx08C Instruction Cache Lines Invalidated

The number of instruction cache lines invalidated.

| UnitMask | Description |
|---|---|
| 7:4 | Reserved. |
| 3 | SMC that hit one or more in-flight instructions |
| 2 | SMC that did not hit any in-flight instructions. |
| 0 | Invalidating probe that did not hit any in-flight instructions. |
| 1 | Invalidating probe that hit one or more in-flight instructions. |

**PMCx099 ITLB Reloads**

The number of ITLB reload requests.

**PMCx09A ITLB Reloads Aborted**

The number of ITLB reloads aborted.

### 3.24.6 Execution Unit Events

See the following events for additional execution unit information:

- PMCx026 [Retired CLFLUSH Instructions].
- PMCx027 [Retired CPUID Instructions].
- PMCx076 [CPU Clocks not Halted].

**PMCx0C0 Retired Instructions**

The number of instructions retired (execution completed and architectural state updated). This count includes exceptions and interrupts - each exception or interrupt is counted as one instruction.

**PMCx0C1 Retired uops**

The number of micro-ops retired. This includes all processor activity (instructions, exceptions, interrupts, microcode assists, etc.).

**PMCx0C2 Retired Branch Instructions**

The number of branch instructions retired. This includes all types of architectural control flow changes, including exceptions and interrupts.

**PMCx0C3 Retired Mispredicted Branch Instructions**

The number of branch instructions retired, of any type, that are not correctly predicted. This includes those for which prediction is not attempted (far control transfers, exceptions and interrupts).

**PMCx0C4 Retired Taken Branch Instructions**

The number of taken branches retired. This includes all types of architectural control flow changes, including exceptions and interrupts.

**PMCx0C5 Retired Taken Branch Instructions Mispredicted**

The number of retired taken branch instructions that are mispredicted.

**PMCx0C6 Retired Far Control Transfers**

The number of far control transfers retired including far call/jump/return, IRET, SYSCALL and SYSRET, plus

exceptions and interrupts. Far control transfers are not subject to branch prediction.

**PMCx0C7 Retired Branch Resyncs**

The number of resync branches. These reflect pipeline restarts due to certain microcode assists and events such as writes to the active instruction stream, among other things. Each occurrence reflects a restart penalty similar to a branch mispredict. This is relatively rare.

**PMCx0C8 Retired Near Returns**

The number of near return instructions (RET or RET Iw) retired.

**PMCx0C9 Retired Near Returns Mispredicted**

The number of near returns retired that are not correctly predicted by the return address predictor. Each such mispredict incurs the same penalty as a mispredicted conditional branch instruction.

**PMCx0CA Retired Indirect Branches Mispredicted**

The number of indirect branch instructions retired where the target address was not correctly predicted.

**PMCx0CB Retired MMX™/FP Instructions**

The number of MMX™, SSE or X87 instructions retired. The UnitMask allows the selection of the individual classes of instructions as given in the table. Each increment represents one complete instruction.

Since this event includes non-numeric instructions it is not suitable for measuring MFLOPS.

| UnitMask | Description |
|---|---|
| 7:3 | Reserved. |
| 2 | SSE and SSE2 instructions |
| 1 | MMX™ and 3DNow!™ instructions |
| 0 | x87 instructions |

**PMCx0CD Interrupts-Masked Cycles**

The number of processor cycles where interrupts are masked (EFLAGS.IF = 0). Using edge-counting with this event gives the number of times IF is cleared; dividing the cycle-count value by this value gives the average length of time that interrupts are disabled on each instance. Compare the edge count with PMCx0CF to determine how often interrupts are disabled for interrupt handling vs. other reasons (e.g. critical sections).

**PMCx0CE Interrupts-Masked Cycles with Interrupt Pending**

The number of processor cycles where interrupts are masked (EFLAGS.IF = 0) and an interrupt is pending. Using edge-counting with this event and comparing the resulting count with the edge count for PMCx0CD gives the proportion of interrupts for which handling is delayed due to prior interrupts being serviced, critical sections, etc. The cycle count value gives the total amount of time for such delays. The cycle count divided by the edge count gives the average length of each such delay.

**PMCx0CF Interrupts Taken**

The number of hardware interrupts taken. This does not include software interrupts (INT n instruction).

**PMCx0D0 Decoder Empty**

The number of processor cycles where the decoder has nothing to dispatch (typically waiting on an instruction

fetch that missed the Icache, or for the target fetch after a branch mispredict).

**PMCx0D1 Dispatch Stalls**

The number of processor cycles where the decoder is stalled for any reason (has one or more instructions ready but can't dispatch them due to resource limitations in execution). This is the combined effect of events D2h - DAh, some of which may overlap; this event reflects the net stall cycles. The more common stall conditions (events D5h, D6h, D7h, D8h, and to a lesser extent D2) may overlap considerably. The occurrence of these stalls is highly dependent on the nature of the code being executed (instruction mix, memory reference patterns, etc.).

**PMCx0D2 Dispatch Stall for Branch Abort to Retire**

The number of processor cycles the decoder is stalled waiting for the pipe to drain after a mispredicted branch. This stall occurs if the corrected target instruction reaches the dispatch stage before the pipe has emptied. See PMCx0D1.

**PMCx0D3 Dispatch Stall for Serialization**

The number of processor cycles the decoder is stalled due to a serializing operation, which waits for the execution pipeline to drain. Relatively rare; mainly associated with system instructions. See PMCx0D1.

**PMCx0D4 Dispatch Stall for Segment Load**

The number of processor cycles the decoder is stalled due to a segment load instruction being encountered while execution of a previous segment load operation is still pending. Relatively rare except in 16-bit code. See PMCx0D1.

**PMCx0D5 Dispatch Stall for Reorder Buffer Full**

The number of processor cycles the decoder is stalled because the reorder buffer is full. May occur simultaneously with certain other stall conditions; see PMCx0D1.

**PMCx0D6 Dispatch Stall for Reservation Station Full**

The number of processor cycles the decoder is stalled because a required integer unit reservation stations is full. May occur simultaneously with certain other stall conditions; see PMCx0D1.

**PMCx0D7 Dispatch Stall for FPU Full**

The number of processor cycles the decoder is stalled because the scheduler for the Floating Point Unit is full. This condition can be caused by a lack of parallelism in FP-intensive code, or by cache misses on FP operand loads (which could also show up as PMCx0D8 instead, depending on the nature of the instruction sequences). May occur simultaneously with certain other stall conditions; see PMCx0D1

**PMCx0D8 Dispatch Stall for LS Full**

The number of processor cycles the decoder is stalled because the Load/Store Unit is full. This generally occurs due to heavy cache miss activity. May occur simultaneously with certain other stall conditions; see PMCx0D1.

**PMCx0D9 Dispatch Stall Waiting for All Quiet**

The number of processor cycles the decoder is stalled waiting for all outstanding requests to the system to be resolved. Relatively rare; associated with certain system instructions and types of interrupts. May partially overlap certain other stall conditions; see PMCx0D1.

**PMCx0DA Dispatch Stall for Far Transfer or Resync to Retire**

The number of processor cycles the decoder is stalled waiting for the execution pipeline to drain before dispatching the target instructions of a far control transfer or a Resync (an instruction stream restart associated with certain microcode assists). Relatively rare; does not overlap with other stall conditions. See PMCx0D1.

### PMCx0DB FPU Exceptions

The number of floating point unit exceptions for microcode assists. The UnitMask may be used to isolate specific types of exceptions.

| UnitMask | Description |
|---|---|
| 7:4 | Reserved. |
| 3 | SSE and x87 microtraps |
| 2 | SSE reclass microfaults |
| 1 | SSE retype microfaults |
| 0 | x87 reclass microfaults |

### PMCx0DC DR0 Breakpoint Matches

The number of matches on the address in breakpoint register DR0, per the breakpoint type specified in DR7. The breakpoint does not have to be enabled. Each instruction breakpoint match incurs an overhead of about 120 cycles; load/store breakpoint matches do not incur any overhead.

### PMCx0DD DR1 Breakpoint Matches

The number of matches on the address in breakpoint register DR1. See notes for PMCx0DC.

### PMCx0DE DR2 Breakpoint Matches

The number of matches on the address in breakpoint register DR2. See notes for PMCx0DC.

### PMCx0DF DR3 Breakpoint Matches

The number of matches on the address in breakpoint register DR3. See notes for PMCx0DC.

### PMCx1C0 Retired x87 Floating Point Operations

The number of x87 floating point ops that have retired.

| UnitMask | Description |
|---|---|
| 7:3 | Reserved. |
| 2 | Divide ops |
| 1 | Multiply ops |
| 0 | Add/subtract ops |

### PMCx1D3 LFENCE Instructions Retired

The number of LFENCE instructions retired.

### PMCx1D4 SFENCE Instructions Retired

The number of SFENCE instructions retired. This counter only counts properly if MSRC001_0015[SlowSfence]=0.

**PMCx1D5 MFENCE Instructions Retired**

The number of MFENCE instructions retired.

### 3.24.7 Memory Controller Events

If more than one UnitMask bit is set for an event, then simultaneous events are counted only once.

**PMCx0E0 DRAM Accesses**

The number of memory accesses performed by the local DRAM controller. The UnitMask may be used to isolate the different DRAM page access cases. Page miss cases incur an extra latency to open a page; page conflict cases incur both a page-close as well as page-open penalties. These penalties may be overlapped by DRAM accesses for other requests and don't necessarily represent lost DRAM bandwidth. The associated penalties are as follows:

Page miss: Trcd (DRAM RAS-to-CAS delay)
Page conflict: Trp + Trcd (DRAM row-precharge time plus RAS-to-CAS delay)

| UnitMask | Description |
|---|---|
| 7 | Read request. |
| 6 | Write request. |
| 5 | DCT1 Page Conflict. |
| 4 | DCT1 Page Miss. |
| 3 | DCT1 Page hit. |
| 2 | DCT0 Page Conflict. |
| 1 | DCT0 Page Miss. |
| 0 | DCT0 Page hit. |

**PMCx0E1 DRAM Controller 0 Page Table Events**

The number of page table events in the local DRAM controller. This table maintains information about which DRAM pages are open. An overflow occurs when a request for a new page arrives when the page table becomes full, as the oldest entry is speculatively closed. Each occurrence reflects an access latency penalty equivalent to a page conflict.

| UnitMask | Description |
|---|---|
| 7:5 | Reserved. |
| 4 | DCT0 Page table is closed due to row inactivity. |
| 3 | DCT0 Page table idle cycle limit decremented. |
| 2 | DCT0 Page table idle cycle limit incremented. |
| 1 | DCT0 Number of stale table entry hits. (hit on a page closed too soon). |
| 0 | DCT0 Page Table Overflow. |

### PMCx0E2 Memory Controller DRAM Command Slots Missed

| UnitMask | Description |
|---|---|
| 7 | DCT1 Prefetch. |
| 6 | DCT0 Prefetch. |
| 5 | DCT1 RBD. |
| 4 | DCT0 RBD. |
| 3:0 | Reserved. |

### PMCx0E3 Memory Controller Turnarounds

The number of turnarounds on the local DRAM data bus. The UnitMask may be used to isolate the different cases. These represent lost DRAM bandwidth, which may be calculated as follows (in bytes per occurrence):

DIMM turnaround: DRAM_width_in_bytes * 2 edges_per_memclk * 2
R/W turnaround: DRAM_width_in_bytes * 2 edges_per_memclk * 1
R/W turnaround: DRAM_width_in_bytes * 2 edges_per_memclk * (Tcl-1)

where DRAM_width_in_bytes is 8 or 16 (for single- or dual-channel systems), and Tcl is the CAS latency of the DRAM in memory system clock cycles (where the memory clock for DDR-400, or PC3200 DIMMS, for example, would be 200 MHz).

| UnitMask | Description |
|---|---|
| 7:5 | Reserved. |
| 4 | DCT1 write-to-read turnaround. |
| 3 | DCT1 read-to-write turnaround. |
| 2 | Reserved. |
| 1 | DCT0 write-to-read turnaround. |
| 0 | DCT0 read-to-write turnaround. |

### PMCx0E4 Memory Controller RBD Queue Events

| UnitMask | Description |
|---|---|
| 7:4 | Reserved. |
| 3 | Bank is closed due to bank conflict with an outstanding request in the RBD queue. |
| 2 | D18F2x[1,0]94[DcqBypassMax] counter reached. |
| 1:0 | Reserved. |

### PMCx0E5 DRAM Controller 1 Page Table Events

The number of page table events in the local DRAM controller. This table maintains information about which DRAM pages are open. An overflow occurs when a request for a new page arrives when the page table becomes full, as the oldest entry is speculatively closed. Each occurrence reflects an access latency penalty equivalent to a page conflict.

| UnitMask | Description |
|---|---|
| 7:5 | Reserved. |
| 4 | DCT1 Page table is closed due to row inactivity. |
| 3 | DCT1 Page table idle cycle limit decremented. |
| 2 | DCT1 Page table idle cycle limit incremented. |
| 1 | DCT1 Number of stale table entry hits. (hit on a page closed too soon). |
| 0 | DCT1 Page Table Overflow. |

**PMCx0E8 Thermal Status**

| UnitMask | Description |
|---|---|
| 7 | PROCHOT_L asserted by an external source and the assertion causes a P-state change. |
| 6 | Number of clocks HTC P-state is active. |
| 5 | Number of clocks HTC P-state is inactive. |
| 4:3 | Reserved. |
| 2 | Number of times the HTC transitions from inactive to active. |
| 1 | Reserved. |
| 0 | MEMHOT_L assertions. |

**PMCx0E9 CPU/IO Requests to Memory/IO**

These events reflect request flow between units, as selected by the UnitMask. It is not possible to tell from these events how much data is going in which direction, as there is no distinction between reads and writes. Also, particularly for IO, the requests may be for varying amounts of data, anywhere from one to sixty-four bytes.

| UnitMask | Description |
|---|---|
| 7:4 | Reserved. |
| 3 | CPU to Mem. |
| 2 | CPU to IO. |
| 1 | IO to Mem. |
| 0 | IO to IO. |

**PMCx0EA Cache Block Commands**

The number of requests made to the system for cache line transfers or coherency state changes, by request type. Each increment represents one cache line transfer, except for Change-to-Dirty. If a Change-to-Dirty request hits on a line in another processor's cache that's in the Owned state, it causes a cache line transfer, otherwise there is no data transfer associated with Change-to-Dirty requests.

| UnitMask | Description |
|---|---|
| 7:6 | Reserved. |
| 5 | Change to Dirty (first store to clean block already in cache). |

| | |
|---|---|
| 4 | Read Block Modified (Dcache store miss refill). |
| 3 | Read Block Shared (Icache refill). |
| 2 | Read Block (Dcache load miss refill). |
| 1 | Reserved. |
| 0 | Victim Block (Writeback). |

### PMCx0EB Sized Commands

The number of Sized Read/Write commands handled by the System Request Interface (local processor and hostbridge interface to the system). These commands may originate from the processor or hostbridge. See PMCx0EC, which provides a separate measure of Hostbridge accesses.

| UnitMask | Description |
|---|---|
| 7:6 | Reserved. |
| 5 | SzRd DW (1-16 dwords). Block-oriented DMA reads, typically cache line size. |
| 4 | SzRd Byte (4 bytes). Legacy or mapped IO. |
| 3 | Posted SzWr DW (1-16 dwords). Block-oriented DMA writes, often cache line sized; also processor Write Combining buffer flushes. |
| 2 | Posted SzWr Byte (1-32 bytes). Sub-cache-line DMA writes, size varies; also flushes of partially-filled Write Combining buffer. |
| 1 | Non-Posted SzWr DW (1-16 dwords). Legacy or mapped IO, typically 1 DWORD. |
| 0 | Non-Posted SzWr Byte (1-32 bytes). Legacy or mapped IO, typically 1-4 bytes. |

### PMCx0EC Probe Responses and Upstream Requests

This covers two unrelated sets of events: cache probe results, and requests received by the hostbridge from devices on a non-coherent link.

**Probe results**: These events reflect the results of probes sent from a memory controller to local caches. They provide an indication of the degree data and code is shared between processors (or moved between processors due to process migration). The dirty-hit events indicate the transfer of a 64-byte cache line to the requester (for a read or cache refill) or the target memory (for a write). The system bandwidth used by these, in terms of bytes per unit of time, may be calculated as 64 times the event count, divided by the elapsed time. Sized writes to memory that cover a full cache line do not incur this cache line transfer -- they simply invalidate the line and are reported as clean hits. Cache line transfers occur for Change2Dirty requests that hit cache lines in the Owned state. (Such cache lines are counted as Modified-state refills for PMCx06C, System Read Responses.)

**Upstream requests**: The upstream read and write events reflect requests originating from a device on a local link. DMA accesses may be anywhere from 1 to 64 bytes in size, but may be dominated by a particular size such as 32 or 64 bytes, depending on the nature of the devices.

| UnitMask | Description |
|---|---|
| 7 | Upstream low priority writes. |
| 6 | Reserved. |
| 5 | Upstream low priority reads. |
| 4 | Upstream high priority reads. |

| | |
|---|---|
| 3 | Probe hit dirty with memory cancel (probed by DMA read or cache refill request). |
| 2 | Probe hit dirty without memory cancel (probed by Sized Write or Change2Dirty). |
| 1 | Probe hit clean. |
| 0 | Probe miss. |

**PMCx0EE DEV Events**

| UnitMask | Description |
|---|---|
| 7 | Reserved. |
| 6 | DEV error |
| 5 | DEV miss |
| 4 | DEV hit |
| 3:0 | Reserved. |

**PMCx1F0 Memory Controller Requests**

Sized Read/Write activity: The Sized Read/Write events reflect 32- or 64-byte transfers (as opposed to other sizes which could be anywhere between 1 and 64 bytes), from either the processor or the Hostbridge. Such accesses from the processor would be due only to write combining buffer flushes, where 32-byte accesses would reflect flushes of partially-filled buffers. Event 65h provides a count of sized write requests associated with WC buffer flushes; comparing that with counts for these events (providing there is very little Hostbridge activity at the same time) give an indication of how efficiently the write combining buffers are being used. Event 65h may also be useful in factoring out WC flushes when comparing these events with the Upstream Requests component of event ECh.

| UnitMask | Description |
|---|---|
| 7 | Reserved. |
| 6 | 64 Byte Sized Reads |
| 5 | 32 Bytes Sized Reads |
| 4 | 64 Bytes Sized Writes |
| 3 | 32 Bytes Sized Writes |
| 2:0 | Reserved. |

### 3.24.8 Crossbar Events

**PMCx1E9 Sideband Signals and Special Cycles**

| UnitMask | Description |
|---|---|
| 7:5 | Reserved. |
| 4 | INVD |
| 3 | WBINVD |
| 2 | SHUTDOWN |

| | |
|---|---|
| 1 | STOPGRANT |
| 0 | Reserved. |

**PMCx1EA Interrupt Events**

| UnitMask | Description |
|---|---|
| 7 | EOI |
| 6 | INT |
| 5 | STARTUP |
| 4 | INIT |
| 3 | NMI |
| 2 | SMI |
| 1 | LPA |
| 0 | Fixed and LPA |

## 4 Register List

The following is a list of all storage elements, context, and registers provided in this document. Page numbers, register mnemonics, and register names are provided.


36 SMMFEC0: SMM IO Trap Offset
37 SMMFEC4: Local SMI Status
37 SMMFEC8: SMM IO Restart Byte
38 SMMFEC9: Auto Halt Restart Offset
38 SMMFECA: NMI Mask
38 SMMFED8: SMM SVM State
39 SMMFEFC: SMM-Revision Identifier
39 SMMFF00: SMM Base Address Register (SMM_BASE)
151 IOCF8: IO-Space Configuration Address Register
151 IOCFC: IO-Space Configuration Data Port
151 D0F0x00: Device/Vendor ID Register
152 D0F0x04: Status/Command Register
152 D0F0x08: Class Code/Revision ID Register
153 D0F0x0C: Header Type Register
153 D0F0x2C: Subsystem and Subvendor ID
153 D0F0x34: Capabilities Pointer Register
153 D0F0x4C: PCI Control Register
154 D0F0x60: Miscellaneous Index Register
154 D0F0x64: Miscellaneous Index Data Register
154 D0F0x64_x00: Northbridge Control
154 D0F0x64_x0B: IOC Link Control
155 D0F0x64_x0C: IOC Bridge Control
155 D0F0x64_x19: Top of Memory 2 Low
155 D0F0x64_x1A: Top of Memory 2 High
156 D0F0x64_x1C: Internal Graphics PCI Control 1
157 D0F0x64_x1D: Internal Graphics PCI Control 2
157 D0F0x64_x20: Device Remap Register
157 D0F0x64_x22: LCLK Control 0
158 D0F0x64_x23: LCLK Control 1
158 D0F0x64_x24: SCLK Control
159 D0F0x64_x46: IOC Features Control
159 D0F0x64_x4D: SMU Request Port
159 D0F0x64_x4E: SMU Read Data
159 D0F0x64_x5[B,9,7,5,3,1]: IOC PCIe Device Control
160 D0F0x64_x6A: Voltage Control
160 D0F0x64_x6B: Voltage Status
161 D0F0x78: Scratch Register
161 D0F0x7C: IOC Configuration Control Register
161 D0F0x84: Link Arbitration
161 D0F0x90: Northbridge Top of Memory
161 D0F0x94: Northbridge ORB Configuration Offset
162 D0F0x98: Northbridge ORB Configuration Data Port
162 D0F0x98_x06: ORB Downstream Control 0
162 D0F0x98_x07: ORB Upstream Arbitration Control 0
162 D0F0x98_x08: ORB Upstream Arbitration Control 1
163 D0F0x98_x09: ORB Upstream Arbitration Control 2
163 D0F0x98_x0C: ORB Upstream Arbitration Control 5
163 D0F0x98_x0E: ORB MSI Interrupt Remap
164 D0F0x98_x1E: ORB Receive Control 0
164 D0F0x98_x28: ORB Transmit Control 0
164 D0F0x98_x2C: ORB Clock Control
165 D0F0x98_x3A: ORB Source Tag Translation Control 2
165 D0F0x98_x4[A:9]: ORB LCLK Clock Control 1-0
165 D0F0x98_x4B: ORB SCLK Clock Control
166 D0F0xE0: Link Index Address
166 D0F0xE4: Link Index Data
167 D0F0xE4_x0[2:1]01_0002: IO Link Hardware Debug
167 D0F0xE4_x0[2:1]01_0010: IO Link Control 1
167 D0F0xE4_x0[2:1]01_0011: IO Link Config Control
167 D0F0xE4_x0[2:1]01_001C: IO Link Control 2
168 D0F0xE4_x0[2:1]01_0020: IO Link Chip Interface Control
168 D0F0xE4_x0[2:1]01_0040: IO Link Phy Control
168 D0F0xE4_x0[2:1]01_00B0: IO Link Strap Link Strap Control
169 D0F0xE4_x0[2:1]01_00C0: IO Link Strap Miscellaneous
169 D0F0xE4_x0[2:1]01_00C1: IO Link Strap Miscellaneous
169 D0F0xE4_x0[2:1]1[2:0]_0010: PIF Control
170 D0F0xE4_x0[2:1]1[2:0]_0011: PIF Pairing
170 D0F0xE4_x0[2:1]1[2:0]_001[3:2]: PIF Power Down Control [1:0]
171 D0F0xE4_x0[2:1]1[2:0]_0015: PIF Transmitter Status
173 D0F0xE4_x0[2:1]2[2:0]_0000: Phy Compensation Control and Calibration Control I
173 D0F0xE4_x0[2:1]2[2:0]_000[2:1]: Phy Impedance Control
174 D0F0xE4_x0[2:1]2[2:0]_000[A:9]: Phy Clock Tree Control
174 D0F0xE4_x0[2:1]2[2:0]_000[C:B]: Phy Serial Bus Packet Control
175 D0F0xE4_x0[2:1]2[2:0]_000D: Phy Serial Bus Compensation Component Packet Enable
175 D0F0xE4_x0[2:1]2[2:0]_2000: Phy PLL Power State Control
176 D0F0xE4_x0[2:1]2[2:0]_2002: Phy PLL Control
176 D0F0xE4_x0[2:1]2[2:0]_2005: Phy PLL Frequency and Mode Control
177 D0F0xE4_x0[2:1]2[2:0]_2008: Phy PLL Update Control
178 D0F0xE4_x0[2:1]2[1:0]_[5:4][7:6,3:0][8,0]1: Phy Receiver DLL Control and Test 1
178 D0F0xE4_x0[2:1]2[1:0]_[5:4][7:6,3:0][8,0]2: Phy Receiver Phase Loop Filter Control
180 D0F0xE4_x0[2:1]2[1:0]_[5:4][7:6,3:0][8,0]5: Phy Receiver Timing Margin Test
181 D0F0xE4_x0[2:1]2[1:0]_[5:4][7:6,3:0][8,0]6: Phy Receiver DFE and DFR Control
181 D0F0xE4_x0[2:1]2[1:0]_[5:4][7:6,3:0][8,0]A: Phy DLL Test and Control 3
183 D0F0xE4_x0[2:1]2[2:0]_[7:6][7:6,3:0][8,0]0: Link Phy Transmit Control
183 D0F0xE4_x0[2:1]2[2:0]_[7:6][7:6,3:0][8,0]5: Phy Transmit Link Configuration
184 D0F0xE4_x0[2:1]2[2:0]_[7:6][7:6,3:0][8,0]6: Phy Transmit Nominal Deemphasis Control
184 D0F0xE4_x013[1:0]_0080: Link Configuration
185 D0F0xE4_x013[1:0]_0[C:8]00: Link Training Control
185 D0F0xE4_x013[1:0]_0[C:8]03: Link Deemphasis Control
185 D0F0xE4_x013[2:0]_8002: IO Link Wrapper Scratch
186 D0F0xE4_x013[2:0]_8011: Link Transmit Clock Gating Control
186 D0F0xE4_x013[2:0]_8012: Idle Resume Clock Gating Control
187 D0F0xE4_x013[2:0]_8013: Transmit Clock Pll Control
188 D0F0xE4_x013[2:0]_8014: Link Transmit Clock Gating Control 2
189 D0F0xE4_x013[2:0]_8015: IO Link IOC Control
189 D0F0xE4_x013[2:0]_8016: Link Clock Switching Control
190 D0F0xE4_x013[2:0]_8021: Transmitter Lane Mux
190 D0F0xE4_x013[2:0]_8022: Receiver Lane Mux
191 D0F0xE4_x013[2:0]_8023: Lane Enable
191 D0F0xE4_x013[2:0]_8025: Lane Mux Power Sequence Control

192 D0F0xE4_x013[1:0]_8031: Lane Counter Status
192 D0F0xE4_x013[2:0]_8060: Soft Reset Command 0
192 D0F0xE4_x013[2:0]_8062: Soft Reset Control 0
193 D0F0xE4_x0130_80F0: BIOS Timer
193 D0F0xE4_x0130_80F1: BIOS Timer Control
193 D0F0xE4_x0130_FFF1: GPP Capabilities
194 D0F0xE4_x013[2:1]_8040: DDI Control
195 D1F0x00: Device/Vendor ID Register
195 D1F0x04: Status/Command Register
196 D1F0x08: Class Code/Revision ID Register
196 D1F0x0C: Header Type Register
196 D1F0x10: Graphic Memory Base Address
196 D1F0x14: Graphics IO Base Address
197 D1F0x14: Graphics Memory Base Address 64
197 D1F0x18: Graphics Memory Mapped Registers Base Address
197 D1F0x1C: Base Address 3
197 D1F0x1C: Graphics Memory Mapped Registers Address 64
197 D1F0x20: Base Address 4
198 D1F0x20: Graphics IO Base Address
198 D1F0x2C: Subsystem and Subvendor ID Register
198 D1F0x30: Expansion ROM Base Address
198 D1F0x34: Capabilities Pointer
198 D1F0x3C: Interrupt Line
199 D1F0x4C: Subsystem and Subvendor ID Mirror
199 D1F0x50: Power Management Capability
199 D1F0x54: Power Management Control and Status
200 D1F0x58: PCI Express Capability
200 D1F0x5C: Device Capability
201 D1F0x60: Device Control and Status
201 D1F0x64: Link Capability
202 D1F0x68: Link Control and Status
203 D1F0x7C: Device Capability 2
203 D1F0x80: Device Control and Status 2
203 D1F0x84: Link Capability 2
203 D1F0x88: Link Control and Status 2
204 D1F0xA0: MSI Capability
204 D1F0xA4: MSI Message Address Low
205 D1F0xA8: MSI Message Address High
205 D1F0xA8: MSI Message Data
205 D1F0xAC: MSI Message Data
205 D1F0x100: Vendor Specific Enhanced Capability
205 D1F0x104: Vendor Specific Header
206 D1F0x108: Vendor Specific 1
206 D1F0x10C: Vendor Specific 2
206 D1F1x00: Device/Vendor ID
206 D1F1x04: Status/Command
207 D1F1x08: Class Code/Revision ID
207 D1F1x0C: Header Type
208 D1F1x10: Audio Registers Base Address
208 D1F1x14: Base Address 1
208 D1F1x18: Base Address 2
208 D1F1x1C: Base Address 3
208 D1F1x20: Base Address 4
208 D1F1x24: Base Address 5
209 D1F1x2C: Subsystem and Subvendor ID
209 D1F1x30: Expansion ROM Base Address
209 D1F1x34: Capabilities Pointer
209 D1F1x3C: Interrupt Line
209 D1F1x4C: Subsystem and Subvendor ID Mirror
209 D1F1x50: Power Management Capability
210 D1F1x54: Power Management Control and Status
210 D1F1x58: PCI Express Capability
211 D1F1x5C: Device Capability
211 D1F1x60: Device Control and Status
212 D1F1x64: Link Capability
212 D1F1x68: Link Control and Status
213 D1F1x7C: Device Capability 2
214 D1F1x80: Device Control and Status 2
214 D1F1x84: Link Capability 2
214 D1F1x88: Link Control and Status 2
215 D1F1xA0: MSI Capability
215 D1F1xA4: MSI Message Address Low
215 D1F1xA8: MSI Message Address High
215 D1F1xA8: MSI Message Data
215 D1F1xAC: MSI Message Data
216 D1F1x100: Vendor Specific Enhanced Capability
216 D1F1x104: Vendor Specific Header
216 D1F1x108: Vendor Specific 1
216 D1F1x10C: Vendor Specific 2
216 D[8:2]F0x00: Device/Vendor ID Register
217 D[8:2]F0x04: Status/Command Register
218 D[8:2]F0x08: Class Code/Revision ID Register
218 D[8:2]F0x0C: Header Type Register
218 D[8:2]F0x18: Bus Number and Secondary Latency Register
218 D[8:2]F0x1C: IO Base and Secondary Status Register
219 D[8:2]F0x20: Memory Limit and Base Register
219 D[8:2]F0x24: Prefetchable Memory Limit and Base Register
219 D[8:2]F0x28: Prefetchable Memory Base High Register
220 D[8:2]F0x2C: Prefetchable Memory Limit High Register
220 D[8:2]F0x30: IO Base and Limit High Register
220 D[8:2]F0x34: Capabilities Pointer Register
220 D[8:2]F0x3C: Bridge Control Register
221 D[8:2]F0x50: Power Management Capability Register
221 D[8:2]F0x54: Power Management Control and Status Register
222 D[8:2]F0x58: PCI Express Capability Register
222 D[8:2]F0x5C: Device Capability Register
223 D[8:2]F0x60: Device Control and Status Register
224 D[8:2]F0x64: IO Link Capability Register
224 D[8:2]F0x68: IO Link Control and Status Register
226 D[8:2]F0x6C: Slot Capability Register
226 D[8:2]F0x70: Slot Control and Status Register
227 D[8:2]F0x74: Root Complex Capability and Control Register
228 D[8:2]F0x78: Root Complex Status Register
228 D[8:2]F0x7C: Device Capability 2
228 D[8:2]F0x80: Device Control and Status 2
229 D[8:2]F0x84: IO Link Capability 2
229 D[8:2]F0x88: IO Link Control and Status 2
230 D[8:2]F0x8C: Slot Capability 2
230 D[8:2]F0x90: Slot Control and Status 2
230 D[8:2]F0xA0: MSI Capability Register
231 D[8:2]F0xA4: MSI Message Address
231 D[8:2]F0xA8: MSI Message Data
231 D[8:2]F0xB0: Subsystem and Subvendor Capability ID Register
231 D[8:2]F0xB4: Subsystem and Subvendor ID Register
231 D[8:2]F0xB8: MSI Capability Mapping
232 D[8:2]F0xBC: MSI Mapping Address Low
232 D[8:2]F0xC0: MSI Mapping Address High
232 D[8:2]F0xE0: Root Port Index

232 D[8:2]F0xE4: Root Port Data
232 D[8:2]F0xE4_x02: Root Port Hardware Debug
232 D[8:2]F0xE4_x20: Root Port TX Control
233 D[8:2]F0xE4_x50: Root Port Lane Status
233 D[8:2]F0xE4_x6A: Root Port Error Control
233 D[8:2]F0xE4_x70: Root Port Receiver Control
233 D[8:2]F0xE4_xA0: Per Port Link Controller (LC) Control
234 D[8:2]F0xE4_xA1: LC Training Control
234 D[8:2]F0xE4_xA2: LC Link Width Control
235 D[8:2]F0xE4_xA3: LC Number of FTS Control
235 D[8:2]F0xE4_xA4: LC Link Speed Control
236 D[8:2]F0xE4_xA5: LC State 0
237 D[8:2]F0xE4_xB1: LC Control 2
237 D[8:2]F0xE4_xB5: LC Control 3
238 D[8:2]F0xE4_xC0: LC Strap Override
238 D[8:2]F0xE4_xC1: Root Port Miscellaneous Strap Override
238 D[8:2]F0x100: Vendor Specific Enhanced Capability Register
238 D[8:2]F0x104: Vendor Specific Header Register
238 D[8:2]F0x108: Vendor Specific 1 Register
239 D[8:2]F0x10C: Vendor Specific 2 Register
239 D[8:2]F0x128: Virtual Channel 0 Resource Status Register
239 D[8:2]F0x150: Advanced Error Reporting Capability
239 D[8:2]F0x154: Uncorrectable Error Status
240 D[8:2]F0x158: Uncorrectable Error Mask
241 D[8:2]F0x15C: Uncorrectable Error Severity
241 D[8:2]F0x160: Correctable Error Status
242 D[8:2]F0x164: Correctable Error Mask
242 D[8:2]F0x168: Advanced Error Control
242 D[8:2]F0x16C: Header Log DW0
243 D[8:2]F0x170: Header Log DW1
243 D[8:2]F0x174: Header Log DW2
243 D[8:2]F0x178: Header Log DW3
243 D[8:2]F0x17C: Root Error Command
243 D[8:2]F0x180: Root Error Status
244 D[8:2]F0x184: Error Source ID
244 D18F0x00: Device/Vendor ID Register
244 D18F0x04: Status/Command Register
245 D18F0x08: Class Code/Revision ID Register
245 D18F0x0C: Header Type Register
245 D18F0x34: Capabilities Pointer Register
245 D18F0x68: Link Transaction Control Register
246 D18F0x6C: Link Initialization Control Register
247 D18F0x168: Extended Link Transaction Control Register
247 D18F1x00: Device/Vendor ID Register
247 D18F1x04: Status/Command Register
247 D18F1x08: Class Code/Revision ID Register
247 D18F1x0C: Header Type Register
248 D18F1x34: Capabilities Pointer Register
248 D18F1x40: DRAM Base Register
248 D18F1x44: DRAM Limit Register
249 D18F1x[B8,B0,A8,A0,98,90,88,80]: Memory Mapped IO Base Registers
249 D18F1x[BC,B4,AC,A4,9C,94,8C,84]: Memory Mapped IO Limit Registers
250 D18F1xC0: IO-Space Base Register
251 D18F1xC4: IO-Space Limit Register
251 D18F1xF0: DRAM Hole Address Register
251 D18F1xF4: VGA Enable Register
252 D18F2x00: Device/Vendor ID Register
252 D18F2x04: Status/Command Register
252 D18F2x08: Class Code/Revision ID Register
253 D18F2x0C: Header Type Register
253 D18F2x34: Capabilities Pointer Register
253 D18F2x[1,0][4C:40]: DRAM CS Base Address Registers
255 D18F2x[1,0][64:60]: DRAM CS Mask Register
255 D18F2x[1,0]78: DRAM Control Register
256 D18F2x[1,0]7C: DRAM Initialization Register
257 D18F2x[1,0]80: DRAM Bank Address Mapping Register
258 D18F2x[1,0]84: DRAM MRS Register
259 D18F2x[1,0]88: DRAM Timing Low Register
260 D18F2x[1,0]8C: DRAM Timing High Register
261 D18F2x[1,0]90: DRAM Configuration Low Register
262 D18F2x[1,0]94: DRAM Configuration High Register
264 D18F2x[1,0]98: DRAM Controller Additional Data Offset Register
265 D18F2x[1,0]9C: DRAM Controller Additional Data Port
265 D18F2x[1,0]9C_x0000_0000: DRAM Output Driver Compensation Control Register
266 D18F2x[1,0]9C_x0000_0[1:0]0[2:1]: DRAM Write Data Timing
267 D18F2x[1,0]9C_x0000_0004: DRAM Address/Command Timing Control Register
268 D18F2x[1,0]9C_x0000_0[1:0]0[6:5]: DRAM Read DQS Timing Control
269 D18F2x[1,0]9C_x0000_0008: DRAM Phy Control Register
271 D18F2x[1,0]9C_x0000_000B: DRAM Phy Status Register
271 D18F2x[1,0]9C_x0000_000C: DRAM Phy Miscellaneous Register
272 D18F2x[1,0]9C_x0000_000D: DRAM Phy DLL Control Register
273 D18F2x[1,0]9C_x0000_00[24:10]: DRAM DQS Receiver Enable Timing Control
274 D18F2x[1,0]9C_x0000_00[44:30]: DRAM DQS Write Timing Control
274 D18F2x[1,0]9C_x0000_00[51:50]: DRAM Phase Recovery Control
275 D18F2x[1,0]9C_x0D0F_0[F,7:0]02: Data Byte Transmit PreDriver Calibration
277 D18F2x[1,0]9C_x0D0F_0[F,7:0]0[A,6]: Data Byte Transmit PreDriver Calibration 2
277 D18F2x[1,0]9C_x0D0F_0[F,7:0]0F: Data Byte DLL Clock Enable
278 D18F2x[1,0]9C_x0D0F_0[F,7:0]10: Data Byte DLL Power Management
278 D18F2x[1,0]9C_x0D0F_0[F,7:0]13: Data Byte DLL Configuration
279 D18F2x[1,0]9C_x0D0F_0[F,7:0]1F: Data Byte Receiver Configuration
280 D18F2x[1,0]9C_x0D0F_0[F,7:0]30: Data Byte DLL Configuration and PowerDown
280 D18F2x[1,0]9C_x0D0F_0[F,7:0]31: Data Byte Fence2 Threshold
281 D18F2x[1,0]9C_x0D0F_2[1:0]02: Clock Transmit PreDriver Calibration
282 D18F2x[1,0]9C_x0D0F_[C,8,2][F,1:0]1F: Receiver Configuration
282 D18F2x[1,0]9C_x0D0F_2[1:0]20: Clock DLL Delay and Configuration
283 D18F2x[1,0]9C_x0D0F_2[1:0]30: Clock Configuration and Power Down
283 D18F2x[1,0]9C_x0D0F_[C,8,2][F,1:0]31: Fence2 Threshold
284 D18F2x[1,0]9C_x0D0F_4009: Cmp Receiver Configuration
284 D18F2x[1,0]9C_x0D0F_[C,8][1:0]02: Transmit PreDriver Calibration
285 D18F2x[1,0]9C_x0D0F_[C,8][1:0][12,0E,0A,06]: Transmit PreDriver Calibration 2
285 D18F2x[1,0]9C_x0D0F_812F: Addr/Cmd Tri-state Configuration
286 D18F2x[1,0]9C_x0D0F_C000: CKE 2.0X Pad Configuration
286 D18F2x[1,0]9C_x0D0F_E003: Phy Calibration Configuration
286 D18F2x[1,0]9C_x0D0F_E006: Phy PLL Lock Time

287 D18F2x[1,0]9C_x0D0F_E00A: Phy Dynamic Power Mode
287 D18F2x[1,0]9C_x0D0F_E013: Phy PLL Regulator Wait Time
287 D18F2x[1,0]A0: DRAM Controller Miscellaneous Register
287 D18F2xA4: DRAM Controller Temperature Throttle Register
288 D18F2x[1,0]A8: DRAM Controller Miscellaneous Register 2
288 D18F2xAC: DRAM Controller Temperature Status Register
288 D18F2x[1,0]F0: DRAM Controller Extra Data Offset Register
289 D18F2x[1,0]F4: DRAM Controller Extra Data Port
289 D18F2x[1,0]F4_x06: DCT Read Timing
290 D18F2x[1,0]F4_x16: DCT Write Timing
290 D18F2x[1,0]F4_x30: DCT Skip Numerator
290 D18F2x[1,0]F4_x31: DCT Skip Denominator
290 D18F2x[1,0]F4_x32: DCT Transmit Fifo Control
291 D18F2x[1,0]F4_x40: DRAM Timing 0
292 D18F2x[1,0]F4_x41: DRAM Timing 1
292 D18F2x[1,0]F4_x83: DCT ODT Control
293 D18F2x[1,0]F4_x180: DCT ODT Control
293 D18F2x[1,0]F4_x182: DCT ODT Control
294 D18F2x[1,0]F4_x200: DCT Power Management
294 D18F2x10C: Interleaved Region Base/Limit Register
295 D18F2x110: DRAM Controller Select Low Register
297 D18F2x114: DRAM Controller Select High Register
297 D18F2x118: Memory Controller Configuration Low Register
298 D18F2x11C: Memory Controller Configuration High Register
300 D18F2x1C0: DRAM Training Control
301 D18F2x1C8: DRAM Training Address Pointer Low
301 D18F2x1CC: DRAM Training Address Pointer High
302 D18F2x1D0: DRAM Write Training Buffer Address
302 D18F2x1D4: DRAM Write Training Data
302 D18F2x[1E0:1D8]: DRAM Training Alternate Address Pointer Low
303 D18F2x1E8: DRAM Training Status
303 D18F3x00: Device/Vendor ID Register
303 D18F3x04: Status/Command Register
303 D18F3x08: Class Code/Revision ID Register
304 D18F3x0C: Header Type Register
304 D18F3x34: Capability Pointer Register
304 D18F3x40: MCA NB Control Register
305 D18F3x44: MCA NB Configuration Register
307 D18F3x48: MCA NB Status Low Register
309 D18F3x4C: MCA NB Status High Register
310 D18F3x50: MCA NB Address Low Register
310 D18F3x54: MCA NB Address High Register
311 D18F3x64: Hardware Thermal Control (HTC) Register
312 D18F3x6C: Upstream Data Buffer Count Register
313 D18F3x74: Upstream Command Buffer Count Register
313 D18F3x7C: In-Flight Queue Buffer Allocation Register
314 D18F3x80: ACPI Power State Control Low
314 D18F3x84: ACPI Power State Control High
314 D18F3x88: NB Configuration Low Registers
315 D18F3x8C: NB Configuration High Registers
315 D18F3xA0: Power Control Miscellaneous Register
316 D18F3xA4: Reported Temperature Control Register
316 D18F3xD4: Clock Power/Timing Control 0 Register
317 D18F3xD8: Clock Power/Timing Control 1 Register
318 D18F3xDC: Clock Power/Timing Control 2 Register
320 D18F3xE4: Thermtrip Status Register
320 D18F3xE8: Northbridge Capabilities Register
321 D18F3xF0: DEV Capability Header Register
322 D18F3xF4: DEV Function/Index Register
322 D18F3xF8: DEV Data Port
322 D18F3xF8_x0: DEV Base Address/Limit Low Register
323 D18F3xF8_x1: DEV Base Address/Limit High Register
323 D18F3xF8_x2: DEV Map Register
324 D18F3xF8_x3: DEV Capabilities Register
324 D18F3xF8_x4: DEV Control Register
325 D18F3xF8_x5: DEV Error Status Register
326 D18F3xF8_x6: DEV Error Address Low Register
326 D18F3xF8_x7: DEV Error Address High Register
326 D18F3xFC: CPUID Family/Model Register
326 D18F3x128: Clock Power/Timing Control 3
327 D18F3x138: Local Hardware Thermal Control (LHTC) Register
328 D18F3x15C: DPM Voltage Control Register
328 D18F3x17C: In-Flight Queue Extended Buffer Allocation Register
328 D18F3x180: Extended NB MCA Configuration Register
329 D18F3x188: NB Extended Configuration Register
329 D18F3x1CC: IBS Control Register
330 D18F3x1E4: SBI Control Register
330 D18F3x1E8: SBI Address Register
331 D18F3x1EC: SBI Data Register
331 D18F3x1F0: Product Information Register
331 D18F4x00: Device/Vendor ID Register
331 D18F4x04: Status/Command Register
332 D18F4x08: Class Code/Revision ID Register
332 D18F4x0C: Header Type Register
332 D18F4x34: Capabilities Pointer Register
332 D18F4x104: TDP Lock Accumulator
333 D18F4x118: C-state Control 1
334 D18F4x11C: C-state Control 2
334 D18F4x120: C-state Policy Control 1
335 D18F4x124: C-state Monitor Control 1
336 D18F4x128: C-state Monitor Control 2
337 D18F4x12C: C6 Base
337 D18F4x134: C-state Monitor Control 3
338 D18F4x138: SMAF Code DID 0
339 D18F4x13C: SMAF Code DID 1
339 D18F4x14C: LPMV Scalar 2
340 D18F4x15C: Core Performance Boost Control
340 D18F4x164: Fixed Errata
340 D18F4x1A4: C-state Monitor Mask
341 D18F4x1A8: CPU State Power Management Dynamic Control 0
343 D18F4x1AC: CPU State Power Management Dynamic Control 1
344 D18F5x00: Device/Vendor ID Register
344 D18F5x04: Status/Command Register
344 D18F5x08: Class Code/Revision ID Register
344 D18F5x0C: Header Type Register
344 D18F5x34: Capabilities Pointer Register
345 D18F6x00: Device/Vendor ID Register
345 D18F6x04: Status/Command Register
345 D18F6x08: Class Code/Revision ID Register
345 D18F6x0C: Header Type Register
345 D18F6x34: Capabilities Pointer Register
345 D18F6x50: Configuration Register Access Control
346 D18F6x[1,0]54: Dram Arbitration Control FEQ Collision
346 D18F6x[1,0]58: Dram Arbitration Control Display Collision
347 D18F6x[1,0]5C: Dram Arbitration Control FEQ Write Protect
347 D18F6x[1,0]60: Dram Arbitration Control Display Write Protect
347 D18F6x[1,0]64: Dram Arbitration Control FEQ Read Protect
348 D18F6x[1,0]68: Dram Arbitration Control Display Read Protect

348 D18F6x[1,0]6C: Dram Arbitration Control FEQ Fairness Timer
348 D18F6x[1,0]70: Dram Arbitration Control Display Fairness Timer
349 D18F6x[1,0]74: Dram Idle Page Close Limit
349 D18F6x78: Dram Prioritization and Arbitration Control
350 D18F6x80: Clocking Control Register 1
350 D18F6x90: NB P-state Config Low
352 D18F6x94: NB P-state Config High
353 D18F6x98: NB P-state Control and Status
353 D18F6x9C: NCLK Reduction Control
354 D18F6x[B8:B0]: Package C-state Residency
354 D18F6xE0: Power Management Residency Counter Enable
355 D18F7x00: Device/Vendor ID Register
355 D18F7x04: Status/Command Register
355 D18F7x08: Class Code/Revision ID Register
355 D18F7x0C: Header Type Register
355 D18F7x34: Capabilities Pointer
356 SMUx01: MCU Config Register
356 SMUx03: MCU IRQ Register
357 SMUx05: SMU Data RAM
357 SMUx0B: SMU RAM Address
357 SMUx0B_x830C: SMU Firmware Version
357 SMUx0B_x8408: RCU Power Gating Sequence 0
358 SMUx0B_x840C: RCU Power Gating Sequence 1
358 SMUx0B_x8410: RCU Power Gating Control
358 SMUx0B_x842C: Alt Vdd Control
359 SMUx0B_x8434: LCLK Scaling Control
359 SMUx0B_x84[48:38:step4]: LCLK DPM Sampling Period [4:0]
359 SMUx0B_x84[54:4C]: LCLK PCIe Up Hysteresis [2:0]
360 SMUx0B_x84[7C:60]: LCLK Activity Thresholds [7:0]
360 SMUx0B_x84[8C:88:step4]: LCLK Scaling Divisor [1:0]
361 SMUx0B_x8490: RCU LCLK Scaling Control 2
361 SMUx0B_x84[9C:94:step4]: RCU LCLK Scaling VID [2:0]
362 SMUx0B_x84A0: RCU Power Gating Control 2
362 SMUx0B_x84A4: RCU Power Gating Control 3
363 SMUx0B_x84A8: RCU Power Gating Control 4
363 SMUx0B_x84[C8:C4]: LCLK PCIe Down Hysteresis [1:0]
363 SMUx0B_x84[D4:D0]: LCLK PCIe Scaling Tolerance [2:1]
364 SMUx0B_x84[E0:D8]: LCLK PCIe Scaling Divisor [2:0]
364 SMUx0B_x84EC: SMU Reset Status
365 SMUx0B_x8580: Power Density Multiplier Control 0
365 SMUx0B_x858C: PCIe Power Gating Arguments 0
365 SMUx0B_x859C: PCIe Power Gating Arguments 2
366 SMUx0B_x85B0: PCIe PLL Wait Time
366 SMUx0B_x85D0: Cable Safe Configuration
366 SMUx0B_x8600: DMA Transaction Array 1
366 SMUx0B_x8604: DMA Transaction Array 2
367 SMUx0B_x8608: DMA Transaction Array 3
367 SMUx0B_x860C: DMA Transaction Array 4
367 SMUx0B_x8610: DMA Transaction Array 5
368 SMUx0B_x8614: DMA Transaction Array 6
368 SMUx0B_x8618: DMA Transaction Array 7
368 SMUx0B_x861C: DMA Transaction Array 8
368 SMUx0B_x8620: DMA Transaction Array 9
369 SMUx0B_x8624: DMA Transaction Array 10
369 SMUx0B_x8628: DMA Transaction Array 11
369 SMUx0B_x862C: DMA Transaction Array 12
370 SMUx0B_x8630: DMA Transaction Array 13
370 SMUx0B_x8634: DMA Transaction Array 14
370 SMUx0B_x8638: DMA Transaction Array 15
370 SMUx0B_x863C: DMA Transaction Array 16
371 SMUx0B_x8640: DMA Transaction Array 17
371 SMUx0B_x86[A0:50:step4]: DMA Scratch Data 21-1
371 SMUx1B: LCLK Deep Sleep Control 0
372 SMUx1D: LCLK Deep Sleep Control 1
372 SMUx33: LCLK Activity Monitor Control
373 SMUx[51:35:step2]: LCLK Activity Monitor Coefficients [14:0]
373 SMUx[5D:55:step2]: PCIe Lane Count Thresholds [4:0]
374 SMUx6F: LCLK Gating Control
374 SMUx71: SCLK Gating Control
375 SMUx73: Clock Gating Control
376 FCRxFE00_4003: VCLK Configuration 0
376 FCRxFE00_4008: VCLK Configuration 1
376 FCRxFE00_4028: VCLK Configuration 2
376 FCRxFE00_4036: Power Configuration Miscellaneous
377 FCRxFE00_6000: NB P-state Configuration 2
377 FCRxFE00_6002: NB P-state Configuration 0
377 FCRxFE00_600E: Clock Configuration
377 FCRxFE00_6022: DCLK/VCLK Selectors
378 FCRxFE00_7006: NB P-state Configuration 1
378 FCRxFE00_7009: NB P-state Configuration 3
378 FCRxFE00_705F: GPU and RC Configuration Miscellaneous 0
379 FCRxFE00_7103: SCLK DPM Configuration 0
379 FCRxFE00_7104: SCLK DPM Configuration 1
379 FCRxFE00_7107: SCLK DPM Configuration 2
379 FCRxFE00_7109: SCLK DPM Configuration 3
380 FCRxFE00_710A: GNB Power Reporting Configuration 0
380 FCRxFE00_710D: Dispclk Configuration
380 FCRxFE00_7110: LCLK DPM Configuration 0
380 FCRxFE00_7113: LCLK DPM Configuration 1
381 FCRxFE00_7114: DCLK Configuration 0
381 FCRxFE00_7117: VCLK/DCLK Configuration 1
381 FCRxFE00_7119: SCLK DPM Configuration 5
381 FCRxFE00_711C: Power Policy Labels
382 FCRxFE00_711E: Power Policy Flags 0
382 FCRxFE00_7121: Power Policy Flags 1
382 FCRxFF30_0134: DCCG Clock Branch Control
382 FCRxFF30_0191: SCLK Power Management Control
383 FCRxFF30_01E4: CG DS Voltage Control
383 FCRxFF30_01F4: Clock Gating Override 0
384 FCRxFF30_01F5: Clock Gating Override 1
385 FCRxFF30_0398: SRBM Soft Reset Generation
386 FCRxFF30_0AE6: GMC Miscellaneous Controls
386 FCRxFF30_1512: GPU PCI Interface Clock Gating Control
386 FCRxFF30_1529: GPU PCI Interface Busy Delay Counter
386 FCRxFF30_1[E,B]7C: Display Controller Front-end (DCFE) Clock Control
386 GMMx00: Memory Mapped Index
386 GMMx04: Memory Mapped Data
387 GMMx100: RCU Indirect Index Register
387 GMMx104: RCU Indirect Data Register
387 GMMx4D0: DCCG Clock Branch Control
387 GMMx670: GPU Control
387 GMMx6[88:84]: SCLK DPM Control [1:0]
388 GMMx690: Activity Monitor Voting
389 GMMx6[CC:94]: SCLK FIR Filter Tap Coefficients
390 GMMx6E0: DPM Control 2
390 GMMx71C: DPM Control 3
391 GMMx720: DPM Control 4

391 GMMx724: DPM Control 6
391 GMMx7[48:38,30:28]: DPM Activity Thresholds 7-0
392 GMMx7[6C:50]: SCLK DPM Activity Sampling Parameters 7-0
392 GMMx770: CG Voltage Control
392 GMMx774: CG Voltage Status
392 GMMx788: DPM Voltage Control
393 GMMx830: SCLK DPM Control 10
394 GMMx15C0: VM L2 Domain Clock Gate Control
394 GMMx2014: GMC Read Group Weight 0
394 GMMx2018: GMC Write Group System
395 GMMx201C: GMC Read Group Weight 2
395 GMMx2020: GMC Write Group Weight
395 GMMx2024: GMC Frame Buffer Location
395 GMMx2028: GMC VM Noncoherent System Memory Top
396 GMMx202C: GMC VM Noncoherent System Memory Bottom
396 GMMx20[C0:B8]: GMC Miscellaneous Domain Clock Gate Control
396 GMMx20D4: GMC Wide Data Path Control
396 GMMx20EC: GMC Read Request Control
396 GMMx2114: GMC Read Request Credits
397 GMMx2144: GMC Read Request Control
397 GMMx21[8C:60]: GMC Read Request Client Control
397 GMMx219[C:0]: GMC Wide Data Path Control
398 GMMx21[D0:A4]: GMC Wide Data Path Client Control
398 GMMx2478: GMC Clock Gating Control
399 GMMx25C0: GMC Master Client Interface Control
399 GMMx25C8: Read Interface Depth
399 GMMx25CC: GMC Write Interface Depth
399 GMMx2620: GMC Local Read Group Weight 0
400 GMMx2624: GMC Local Write Group Weight 0
400 GMMx2628: GMC Local Read Group Weight 1
400 GMMx262C: GMC Local Write Group Weight 1
401 GMMx2630: GMC External Read Group Weight
401 GMMx2634: GMC External Write Group Weight
401 GMMx26[58:50]: GMC Clock Gating Control
401 GMMx2[8D8,77C]: GMC DRAM Timing
402 GMMx2[8DC,780]: GMC DRAM Timing 2
402 GMMx27[88:84]: Weight Manager Control
402 GMMx27[A0:9C]: GMC Group 3-0 Read and Write Arbitration Timer Control
403 GMMx27[D0:CC]: List Manager Control
403 GMMx27DC: GMC Read Arbitration Credit Control
403 GMMx27E0: GMC Write Arbitration Credit Control
403 GMMx2814: UVD Read Latency Control
403 GMMx28[38:1C]: GMC DCT CS Base Address
404 GMMx28[48:3C]: GMC DCT[1:0] CS[3:0] Mask
404 GMMx28[50:4C]: GMC DCT[1:0] Bank Address Mapping
405 GMMx2854: GMC DRAM Control
405 GMMx2858: GMC DRAM Control 2
406 GMMx285C: GMC DRAM Hole Address
406 GMMx2860: GMC DRAM Interleave Region
406 GMMx2864: DRAM Address Swizzle0
407 GMMx2868: DRAM Address Swizzle 1
407 GMMx28[78:6C]: GMC DRAM Aperture Base 3-0
407 GMMx28[88:7C]: GMC DRAM Aperture Limit 3-0
407 GMMx288C: GMC C6 Save Base
408 GMMx2890: GMC C6 Save Limit
408 GMMx2894: GMC DRAM Aperture Default Base Address
408 GMMx2898: GMC Frame Buffer Offset
408 GMMx28C8: GMC IFQ Arbitration
408 GMMx28EC: GMC DCT Interface Control
409 GMMx2B8C: GMC Register Engine RAM Index
409 GMMx2B90: GMC Register Engine RAM Data
409 GMMx2B94: GMC RENG Execution Control
409 GMMx2B98: GMC Miscellaneous Controls
410 GMMx2C04: HDP Default Surface Base Address
410 GMMx5428: Configuration Memory Size
410 GMMx5490: Frame Buffer Access Control
410 GMMx6[3FC,12C:110]: Display Output Scratch
410 GMMx[79,6D]F0: Display Controller Front-end (DCFE) Clock Control
411 APIC20: APIC ID Register
411 APIC30: APIC Version Register
411 APIC80: Task Priority Register
411 APIC90: Arbitration Priority Register
412 APICA0: Processor Priority Register
412 APICB0: End of Interrupt Register
412 APICC0: Remote Read Register
412 APICD0: Logical Destination Register
412 APICE0: Destination Format Register
413 APICF0: Spurious Interrupt Vector Register
413 APIC[170:100]: In-Service Registers
413 APIC[1F0:180]: Trigger Mode Registers
414 APIC[270:200]: Interrupt Request Registers
414 APIC280: Error Status Register
415 APIC300: Interrupt Command Register Low
416 APIC310: Interrupt Command Register High
416 APIC320: Timer Local Vector Table Entry
416 APIC330: Thermal Local Vector Table Entry
417 APIC340: Performance Counter Vector Table Entry
417 APIC350: Local Interrupt 0 (Legacy INTR) Local Vector Table Entry
418 APIC360: Local Interrupt 1(Legacy NMI) Local Vector Table Entry
418 APIC370: Error Local Vector Table Entry
418 APIC380: Timer Initial Count Register
418 APIC390: Timer Current Count Register
418 APIC3E0: Timer Divide Configuration
419 APIC400: Extended APIC Feature Register
419 APIC410: Extended APIC Control Register
420 APIC420: Specific End Of Interrupt Register
420 APIC[4F0:480]: Interrupt Enable
420 APIC[530:500]: Extended Interrupt [3:0] Local Vector Table Registers
421 CPUID Fn0000_0000_EAX: Processor Vendor and Largest Standard Function Number
421 CPUID Fn0000_0000_E[B,C,D]X: Processor Vendor and Largest Standard Function Number
421 CPUID Fn0000_0001_EAX: Family, Model, Stepping Identifiers
422 CPUID Fn0000_0001_EBX: LocalApicId, LogicalProcessorCount, CLFlush, 8BitBrandId
422 CPUID Fn0000_0001_ECX: Feature Identifiers
422 CPUID Fn0000_0001_EDX: Feature Identifiers
423 CPUID Fn0000_000[4,3,2]: Reserved
423 CPUID Fn0000_0005_EAX: Monitor/MWait
423 CPUID Fn0000_0005_EBX: Monitor/MWait
424 CPUID Fn0000_0005_ECX: Monitor/MWait
424 CPUID Fn0000_0005_EDX: Monitor/MWait
424 CPUID Fn0000_0006_EAX: Power Management Features
424 CPUID Fn0000_0006_EBX: Power Management Features
424 CPUID Fn0000_0006_ECX: Power Management Features
424 CPUID Fn0000_0006_EDX: Power Management Features

424 CPUID Fn8000_0000_EAX: Processor Vendor and Largest Extended Function Number
425 CPUID Fn8000_0000_E[B,C,D]X: Processor Vendor and Largest Extended Function Number
425 CPUID Fn8000_0001_EAX: Family, Model, Stepping Identifiers
425 CPUID Fn8000_0001_EBX: BrandId Identifier
425 CPUID Fn8000_0001_ECX: Feature Identifiers
426 CPUID Fn8000_0001_EDX: Feature Identifiers
427 CPUID Fn8000_000[4,3,2]_E[D,C,B,A]X: Processor Name String Identifier
427 CPUID Fn8000_0005_EAX: L1 TLB Identifiers
428 CPUID Fn8000_0005_EBX: L1 TLB Identifiers
428 CPUID Fn8000_0005_ECX: L1 Cache Identifiers
428 CPUID Fn8000_0005_EDX: L1 Cache Identifiers
428 CPUID Fn8000_0006_EAX: L2 Cache and L2 TLB Identifiers
429 CPUID Fn8000_0006_EBX: L2 Cache and L2 TLB Identifiers
429 CPUID Fn8000_0006_ECX: L2 Cache and L2 TLB Identifiers
429 CPUID Fn8000_0006_EDX: Reserved
429 CPUID Fn8000_0007_E[A,B,C]X: Advanced Power Management Information
430 CPUID Fn8000_0007_EDX: Advanced Power Management Information
430 CPUID Fn8000_0008_EAX: Long Mode Address Size Identifiers
430 CPUID Fn8000_0008_EBX: Long Mode Address Size Identifiers
430 CPUID Fn8000_0008_ECX: Long Mode Address Size Identifiers
431 CPUID Fn8000_0008_EDX: Long Mode Address Size Identifiers
431 CPUID Fn8000_0009: Reserved
431 CPUID Fn8000_000A_EAX: SVM Revision and Feature Identification
431 CPUID Fn8000_000A_EBX: SVM Revision and Feature Identification
431 CPUID Fn8000_000A_ECX: SVM Revision and Feature Identification
432 CPUID Fn8000_000A_EDX: SVM Revision and Feature Identification
432 CPUID Fn8000_00[18:0B]: Reserved
432 CPUID Fn8000_0019_EAX: TLB 1GB Page Identifiers
432 CPUID Fn8000_0019_EBX: TLB 1GB Page Identifiers
433 CPUID Fn8000_0019_E[C,D]X: TLB 1GB Page Identifiers
433 CPUID Fn8000_001A_EAX: Performance Optimization Identifiers
433 CPUID Fn8000_001A_E[B,C,D]X: Performance Optimization Identifiers
433 CPUID Fn8000_001B_EAX: Instruction Based Sampling Identifiers
433 CPUID Fn8000_001B_E[B,C,D]X: Instruction Based Sampling Identifiers
434 MSR0000_0000: Load-Store MCA Address Register
434 MSR0000_0001: Load-Store MCA Status Register
434 MSR0000_0010: Time Stamp Counter Register (TSC)
434 MSR0000_001B: APIC Base Address Register (APIC_BAR)
434 MSR0000_002A: Cluster ID Register (EBL_CR_POWERON)
435 MSR0000_00E7: Max Performance Frequency Clock Count (MPERF)
435 MSR0000_00E8: Actual Performance Frequency Clock Count (APERF)
435 MSR0000_00FE: MTRR Capabilities Register (MTRRcap)
435 MSR0000_0174: SYSENTER CS Register (SYSENTER_CS)
436 MSR0000_0175: SYSENTER ESP Register (SYSENTER_ESP)
436 MSR0000_0176: SYSENTER EIP Register (SYSENTER_EIP)
436 MSR0000_0179: Global Machine Check Capabilities Register (MCG_CAP)
436 MSR0000_017A: Global Machine Check Status Register (MCG_STAT)
437 MSR0000_017B: Global Machine Check Exception Reporting Control Register (MCG_CTL)
437 MSR0000_01D9: Debug Control Register (DBG_CTL_MSR)
437 MSR0000_01DB: Last Branch From IP Register (BR_FROM)
438 MSR0000_01DC: Last Branch To IP Register (BR_TO)
438 MSR0000_01DD: Last Exception From IP Register
438 MSR0000_01DE: Last Exception To IP Register
438 MSR0000_020[E,C,A,8,6,4,2,0]: Variable-Size MTRRs (MTRRphysBasen)
439 MSR0000_020[F,D,B,9,7,5,3,1]: Variable-Size MTRRs (MTRRphysMaskn)
439 MSR0000_02[6F:68,59,58,50]: Fixed-Size MTRRs
441 MSR0000_0277: Page Attribute Table Register (PAT)
441 MSR0000_02FF: MTRR Default Memory Type Register (MTRRdefType)
442 MSR0000_0400: DC Machine Check Control Register (MC0_CTL)
442 MSR0000_0401: DC Machine Check Status Register (MC0_STATUS)
444 MSR0000_0402: DC Machine Check Address Register (MC0_ADDR)
445 MSR0000_0403: DC Machine Check Miscellaneous Register (MC0_MISC)
445 MSR0000_0404: IC Machine Check Control Register (MC1_CTL)
446 MSR0000_0405: IC Machine Check Status Register (MC1_STATUS)
447 MSR0000_0406: IC Machine Check Address Register (MC1_ADDR)
448 MSR0000_0407: IC Machine Check Miscellaneous Register (MC1_MISC)
448 MSR0000_0408: BU Machine Check Control Register (MC2_CTL)
449 MSR0000_0409: BU Machine Check Status Register (MC2_STATUS)
450 MSR0000_040A: BU Machine Check Address Register (MC2_ADDR)
450 MSR0000_040B: BU Machine Check Miscellaneous Register (MC2_MISC)
450 MSR0000_040C: LS Machine Check Control Register (MC3_CTL)
451 MSR0000_040D: LS Machine Check Status Register (MC3_STATUS)
451 MSR0000_040E: LS Machine Check Address Register (MC3_ADDR)
451 MSR0000_040F: LS Machine Check Miscellaneous Register (MC3_MISC)
451 MSR0000_0410: NB Machine Check Control Register (MC4_CTL)
452 MSR0000_0411: NB Machine Check Status Register (MC4_STATUS)
452 MSR0000_0412: NB Machine Check Address Register (MC4_ADDR)
452 MSR0000_0413: Reserved
452 MSR0000_0414: FR Machine Check Control Register (MC5_CTL)
452 MSR0000_0415: FR Machine Check Status Register (MC5_STATUS)
453 MSR0000_0416: FR Machine Check Address Register (MC5_ADDR)
453 MSR0000_0417: FR Machine Check Miscellaneous Register (MC5_MISC)
453 MSRC000_0080: Extended Feature Enable Register (EFER)
454 MSRC000_0081: SYSCALL Target Address Register (STAR)
454 MSRC000_0082: Long Mode SYSCALL Target Address Register (STAR64)
454 MSRC000_0083: Compatibility Mode SYSCALL Target Address Register (STARCOMPAT)
454 MSRC000_0084: SYSCALL Flag Mask Register (SYSCALL_FLAG_MASK)
455 MSRC000_0100: FS Base Register (FS_BASE)
455 MSRC000_0101: GS Base Register (GS_BASE)
455 MSRC000_0102: Kernel GS Base Register (KernelGSbase)
455 MSRC000_0103: Auxiliary Time Stamp Counter Register (TSC_AUX)
455 MSRC001_00[03:00]: Performance Event Select Register (PERF_CTL[3:0])
457 MSRC001_00[07:04]: Performance Event Counter Registers (PERF_CTR[3:0])
458 MSRC001_0010: System Configuration Register (SYS_CFG)

459 MSRC001_0015: Hardware Configuration Register (HWCR)
460 MSRC001_00[18,16]: IO Range Registers Base (IORR_BASE[1:0])
461 MSRC001_00[19,17]: IO Range Registers Mask (IORR_MASK[1:0])
461 MSRC001_001A: Top Of Memory Register (TOP_MEM)
461 MSRC001_001D: Top Of Memory 2 Register (TOM2)
461 MSRC001_001F: Northbridge Configuration Register (NB_CFG)
462 MSRC001_0022: Machine Check Exception Redirection Register
462 MSRC001_00[35:30]: Processor Name String Registers
462 MSRC001_0044: DC Machine Check Control Mask (MC0_CTL_MASK)
463 MSRC001_0045: IC Machine Check Control Mask (MC1_CTL_MASK)
463 MSRC001_0046: BU Machine Check Control Mask (MC2_CTL_MASK)
463 MSRC001_0047: LS Machine Check Control Mask (MC3_CTL_MASK)
464 MSRC001_0048: NB Machine Check Control Mask (MC4_CTL_MASK)
464 MSRC001_0049: FR Machine Check Control Mask (MC5_CTL_MASK)
464 MSRC001_00[53:50]: IO Trap Registers (SMI_ON_IO_TRAP_[3:0])
465 MSRC001_0054: IO Trap Control Register (SMI_ON_IO_TRAP_CTL_STS)
466 MSRC001_0055: Interrupt Pending
466 MSRC001_0056: SMI Trigger IO Cycle Register
466 MSRC001_0057: Reserved
466 MSRC001_0058: MMIO Configuration Base Address
467 MSRC001_0060: BIST Results Register
468 MSRC001_0061: P-State Current Limit Register
468 MSRC001_0062: P-State Control Register
468 MSRC001_0063: P-State Status Register
468 MSRC001_00[6B:64]: P-State [7:0] Registers
469 MSRC001_0071: COFVID Status Register
471 MSRC001_0073: C-state Address Register
471 MSRC001_0074: CPU Watchdog Timer Register (CpuWdtCfg)
472 MSRC001_0111: SMM Base Address Register (SMM_BASE)
472 MSRC001_0112: SMM TSeg Base Address Register (SMMAddr)
473 MSRC001_0113: SMM TSeg Mask Register (SMMMask)
474 MSRC001_0114: Virtual Machine Control Register (VM_CR)
474 MSRC001_0115: IGNNE Register (IGNNE)
475 MSRC001_0116: SMM Control Register (SMM_CTL)
475 MSRC001_0117: Virtual Machine Host Save Physical Address Register (VM_HSAVE_PA)
475 MSRC001_0118: SVM Lock Key
476 MSRC001_0119: SMM Lock Key
476 MSRC001_011A: Local SMI Status
476 MSRC001_0140: OS Visible Work-around MSR0 (OSVW_ID_Length)
476 MSRC001_0141: OS Visible Work-around MSR1 (OSVW Status)
476 MSRC001_1004: CPUID Features Register (Features)
477 MSRC001_1005: Extended CPUID Features Register (ExtFeatures)
477 MSRC001_1020: Load-Store Configuration Register (LS_CFG)
477 MSRC001_1021: Instruction Cache Configuration Register (IC_CFG)
477 MSRC001_1022: Data Cache Configuration Register (DC_CFG)
478 MSRC001_1029: Decode Configuration Register (DE_CFG)
478 MSRC001_102A: Bus Unit Configuration 2 Register (BU_CFG2)
478 MSRC001_1030: IBS Fetch Control Register (IbsFetchCtl)
479 MSRC001_1031: IBS Fetch Linear Address Register (IbsFetchLinAd)
480 MSRC001_1032: IBS Fetch Physical Address Register (IbsFetchPhysAd)
480 MSRC001_1033: IBS Execution Control Register (IbsOpCtl)
481 MSRC001_1034: IBS Op Logical Address Register (IbsOpRip)
481 MSRC001_1035: IBS Op Data Register (IbsOpData)
481 MSRC001_1036: IBS Op Data 2 Register (IbsOpData2)
482 MSRC001_1037: IBS Op Data 3 Register (IbsOpData3)
483 MSRC001_1038: IBS DC Linear Address Register (IbsDcLinAd)
483 MSRC001_1039: IBS DC Physical Address Register (IbsDcPhysAd)
483 MSRC001_103A: IBS Control Register
484 MSRC001_103B: IBS Branch Target Address
484 PMCx000: Dispatched FPU Operations
484 PMCx001: Cycles in which the FPU is Empty
485 PMCx002: Dispatched Fast Flag FPU Operations
485 PMCx003: Retired SSE Operations
485 PMCx004: Retired Move Ops
485 PMCx005: Retired Serializing Ops
486 PMCx006: Number of Cycles that a Serializing uop is in the FP Scheduler
486 PMCx020: Segment Register Loads
486 PMCx021: Pipeline Restart Due to Self-Modifying Code
486 PMCx022: Pipeline Restart Due to Probe Hit
486 PMCx023: LS Buffer 2 Full
487 PMCx024: Locked Operations
487 PMCx026: Retired CLFLUSH Instructions
487 PMCx027: Retired CPUID Instructions
487 PMCx02A: Cancelled Store to Load Forward Operations
487 PMCx02B: SMIs Received
487 PMCx040: Data Cache Accesses
487 PMCx041: Data Cache Misses
487 PMCx042: Data Cache Refills from L2 or Northbridge
488 PMCx043: Data Cache Refills from the northbridge
488 PMCx044: Data Cache Lines Evicted
489 PMCx045: L1 DTLB Miss and L2 DTLB Hit
489 PMCx046: L1 DTLB and L2 DTLB Miss
489 PMCx047: Misaligned Accesses
489 PMCx048: Microarchitectural Late Cancel of an Access
489 PMCx049: Microarchitectural Early Cancel of an Access
489 PMCx04B: Prefetch Instructions Dispatched
490 PMCx04C: DCACHE Misses by Locked Instructions
490 PMCx04D: L1 DTLB Hit
490 PMCx052: Ineffective Software Prefetches
490 PMCx054: Global TLB Flushes
490 PMCx065: Memory Requests by Type
491 PMCx067: Data Prefetcher
491 PMCx06C: Northbridge Read Responses by Coherency State
492 PMCx06D: Octwords Written to System
492 PMCx076: CPU Clocks not Halted
492 PMCx07D: Requests to L2 Cache
492 PMCx07E: L2 Cache Misses
493 PMCx07F: L2 Fill/Writeback
493 PMCx165: Page Size Mismatches
493 PMCx080: Instruction Cache Fetches
493 PMCx081: Instruction Cache Misses
493 PMCx082: Instruction Cache Refills from L2
494 PMCx083: Instruction Cache Refills from System
494 PMCx084: L1 ITLB Miss, L2 ITLB Hit
494 PMCx085: L1 ITLB Miss, L2 ITLB Miss
494 PMCx086: Pipeline Restart Due to Instruction Stream Probe

494 PMCx087: Instruction Fetch Stall
494 PMCx088: Return Stack Hits
494 PMCx089: Return Stack Overflows
494 PMCx08B: Instruction Cache Victims
494 PMCx08C: Instruction Cache Lines Invalidated
495 PMCx099: ITLB Reloads
495 PMCx09A: ITLB Reloads Aborted
495 PMCx0C0: Retired Instructions
495 PMCx0C1: Retired uops
495 PMCx0C2: Retired Branch Instructions
495 PMCx0C3: Retired Mispredicted Branch Instructions
495 PMCx0C4: Retired Taken Branch Instructions
495 PMCx0C5: Retired Taken Branch Instructions Mispredicted
495 PMCx0C6: Retired Far Control Transfers
496 PMCx0C7: Retired Branch Resyncs
496 PMCx0C8: Retired Near Returns
496 PMCx0C9: Retired Near Returns Mispredicted
496 PMCx0CA: Retired Indirect Branches Mispredicted
496 PMCx0CB: Retired MMX™/FP Instructions
496 PMCx0CD: Interrupts-Masked Cycles
496 PMCx0CE: Interrupts-Masked Cycles with Interrupt Pending
496 PMCx0CF: Interrupts Taken
496 PMCx0D0: Decoder Empty
497 PMCx0D1: Dispatch Stalls
497 PMCx0D2: Dispatch Stall for Branch Abort to Retire
497 PMCx0D3: Dispatch Stall for Serialization
497 PMCx0D4: Dispatch Stall for Segment Load
497 PMCx0D5: Dispatch Stall for Reorder Buffer Full
497 PMCx0D6: Dispatch Stall for Reservation Station Full
497 PMCx0D7: Dispatch Stall for FPU Full
497 PMCx0D8: Dispatch Stall for LS Full
497 PMCx0D9: Dispatch Stall Waiting for All Quiet
497 PMCx0DA: Dispatch Stall for Far Transfer or Resync to Retire
498 PMCx0DB: FPU Exceptions
498 PMCx0DC: DR0 Breakpoint Matches
498 PMCx0DD: DR1 Breakpoint Matches
498 PMCx0DE: DR2 Breakpoint Matches
498 PMCx0DF: DR3 Breakpoint Matches
498 PMCx1C0: Retired x87 Floating Point Operations
498 PMCx1D3: LFENCE Instructions Retired
498 PMCx1D4: SFENCE Instructions Retired
499 PMCx1D5: MFENCE Instructions Retired
499 PMCx0E0: DRAM Accesses
499 PMCx0E1: DRAM Controller 0 Page Table Events
500 PMCx0E2: Memory Controller DRAM Command Slots Missed
500 PMCx0E3: Memory Controller Turnarounds
500 PMCx0E4: Memory Controller RBD Queue Events
500 PMCx0E5: DRAM Controller 1 Page Table Events
501 PMCx0E8: Thermal Status
501 PMCx0E9: CPU/IO Requests to Memory/IO
501 PMCx0EA: Cache Block Commands
502 PMCx0EB: Sized Commands
502 PMCx0EC: Probe Responses and Upstream Requests
503 PMCx0EE: DEV Events
503 PMCx1F0: Memory Controller Requests
503 PMCx1E9: Sideband Signals and Special Cycles
504 PMCx1EA: Interrupt Events