

**BIOS and Kernel  
Developer's Guide  
(BKDG)  
for AMD Family 16h  
Models 00h-0Fh  
Processors**

© 2013-2015 Advanced Micro Devices, Inc. All rights reserved.

The information contained herein is for informational purposes only, and is subject to change without notice. While every precaution has been taken in the preparation of this document, it may contain technical inaccuracies, omissions and typographical errors, and AMD is under no obligation to update or otherwise correct this information. Advanced Micro Devices, Inc. makes no representations or warranties with respect to the accuracy or completeness of the contents of this document, and assumes no liability of any kind, including the implied warranties of noninfringement, merchantability or fitness for particular purposes, with respect to the operation or use of AMD hardware, software or other products described herein. No license, including implied or arising by estoppel, to any intellectual property rights is granted by this document. Terms and limitations applicable to the purchase or use of AMD's products are as set forth in a signed agreement between the parties or in AMD's Standard Terms and Conditions of Sale.

### **Trademarks**

AMD, the AMD Arrow logo, and combinations thereof are trademarks of Advanced Micro Devices, Inc. Other product names used in this publication are for identification purposes only and may be trademarks of their respective companies.

HyperTransport is a licensed trademark of the HyperTransport Technology Consortium.

Microsoft and Windows are registered trademarks of Microsoft Corporation.

MMX is a trademark of Intel Corporation.

PCI Express and PCIe are registered trademarks of PCI-Special Interest Group (PCI-SIG).

### **Dolby Laboratories, Inc.**

Manufactured under license from Dolby Laboratories.

### **Rovi Corporation**

This device is protected by U.S. patents and other intellectual property rights. The use of Rovi Corporation's copy protection technology in the device must be authorized by Rovi Corporation and is intended for home and other limited pay-per-view uses only, unless otherwise authorized in writing by Rovi Corporation.

Reverse engineering or disassembly is prohibited.

USE OF THIS PRODUCT IN ANY MANNER THAT COMPLIES WITH THE MPEG ACTUAL OR DE FACTO VIDEO AND/OR AUDIO STANDARDS IS EXPRESSLY PROHIBITED WITHOUT ALL NECESSARY LICENSES UNDER APPLICABLE PATENTS. SUCH LICENSES MAY BE ACQUIRED FROM VARIOUS THIRD PARTIES INCLUDING, BUT NOT LIMITED TO, IN THE MPEG PATENT PORTFOLIO, WHICH LICENSE IS AVAILABLE FROM MPEG LA, L.L.C., 6312 S. FIDDLERS GREEN CIRCLE, SUITE 400E, GREENWOOD VILLAGE, COLORADO 80111.

# Table of Contents

<b>1</b>	<b>Overview</b>	<b>20</b>
1.1	Intended Audience	20
1.2	Reference Documents	20
1.3	Conventions	20
1.3.1	Numbering	20
1.3.2	Arithmetic And Logical Operators	21
1.3.3	Operator Precedence and Associativity	22
1.4	Definitions	22
1.5	Changes Between Revisions and Product Variations	27
1.5.1	Revision Conventions	27
1.5.2	Major Changes	28
1.5.2.1	Major Changes to Core/NB Performance Counters	28
<b>2</b>	<b>Functional Description</b>	<b>28</b>
2.1	Processor Overview	28
2.2	System Overview	30
2.3	Processor Initialization	30
2.3.1	BSC Initialization	30
2.3.2	AP Initialization	31
2.3.3	Using L2 Cache as General Storage During Boot	31
2.3.4	Instruction Cache Configuration Register Usage Requirements	33
2.4	Core	33
2.4.1	L2 complex	33
2.4.2	Caches and TLBs	33
2.4.2.1	Registers Shared by Cores in a L2 complex	34
2.4.3	Virtual Address Space	34
2.4.4	Processor Cores and Downcoring	34
2.4.4.1	Software Downcoring using D18F3x190[DisCore]	35
2.4.5	Physical Address Space	35
2.4.6	System Address Map	35
2.4.6.1	Memory Access to the Physical Address Space	35
2.4.6.1.1	Determining Memory Type	36
2.4.6.1.2	Determining The Access Destination for Core Accesses	36
2.4.7	Timers	36
2.4.8	Interrupts	37
2.4.8.1	Local APIC	37
2.4.8.1.1	Detecting and Enabling	37
2.4.8.1.2	APIC Register Space	37
2.4.8.1.3	ApicId Enumeration Requirements	37
2.4.8.1.4	Physical Destination Mode	38
2.4.8.1.5	Logical Destination Mode	38
2.4.8.1.6	Interrupt Delivery	38
2.4.8.1.7	Vectored Interrupt Handling	39
2.4.8.1.8	Interrupt Masking	39
2.4.8.1.9	Spurious Interrupts	39
2.4.8.1.10	Spurious Interrupts Caused by Timer Tick Interrupt	39
2.4.8.1.11	Lowest-Priority Interrupt Arbitration	40
2.4.8.1.12	Inter-Processor Interrupts	40

2.4.8.1.13	APIC Timer Operation	40
2.4.8.1.14	Generalized Local Vector Table	40
2.4.8.1.15	State at Reset	41
2.4.8.2	System Management Mode (SMM)	41
2.4.8.2.1	SMM Overview	41
2.4.8.2.2	Operating Mode and Default Register Values	41
2.4.8.2.3	SMI Sources And Delivery	42
2.4.8.2.4	SMM Initial State	42
2.4.8.2.5	SMM Save State	43
2.4.8.2.6	Exceptions and Interrupts in SMM	48
2.4.8.2.7	The Protected ASeg and TSeg Areas	48
2.4.8.2.8	SMM Special Cycles	48
2.4.8.2.9	Locking SMM	48
2.4.8.2.10	Synchronizing SMM Entry (Spring-Boarding)	48
2.4.9	Secure Virtual Machine Mode (SVM)	50
2.4.9.1	BIOS support for SVM Disable	50
2.4.10	CPUID Instruction	51
2.4.10.1	Multi-Core Support	51
2.5	Power Management	52
2.5.1	Processor Power Planes And Voltage Control	52
2.5.1.1	Serial VID Interface	52
2.5.1.1.1	SVI2 Features	52
2.5.1.2	Internal VID Registers and Encodings	53
2.5.1.2.1	MinVid and MaxVid Check	53
2.5.1.3	Low Power Features	53
2.5.1.3.1	PSIx_L Bit	53
2.5.1.3.1.1	BIOS Requirements for PSIO_L	53
2.5.1.3.2	Low Power Voltages	55
2.5.1.4	Voltage Transitions	55
2.5.1.4.1	Hardware-Initiated Voltage Transitions	55
2.5.1.4.2	Software-Initiated Voltage Transitions	55
2.5.1.4.2.1	Software-Initiated NB Voltage Transitions	55
2.5.2	Frequency and Voltage Domain Dependencies	56
2.5.2.1	Dependencies Between Cores	56
2.5.2.2	Dependencies Between Subcomponents on VDDNB	56
2.5.2.3	BIOS Requirements for Power Plane Initialization	56
2.5.3	CPU Power Management	56
2.5.3.1	Core P-states	56
2.5.3.1.1	Core P-state Naming and Numbering	57
2.5.3.1.1.1	Software P-state Numbering	57
2.5.3.1.1.2	Hardware P-state Numbering	58
2.5.3.1.2	Core P-state Control	58
2.5.3.1.3	Core P-state Visibility	59
2.5.3.1.4	Core P-state Limits	59
2.5.3.1.5	Core P-state Transition Behavior	60
2.5.3.1.6	BIOS Requirements for Core P-state Initialization and Transitions	60
2.5.3.1.7	Processor-Systemboard Power Delivery Compatibility Check	61
2.5.3.1.8	BIOS COF and VID Requirements After Warm Reset	62
2.5.3.1.8.1	Core Maximum P-state Transition Sequence After Warm Reset	63
2.5.3.1.8.2	Core Minimum P-state Transition Sequence After Warm Reset	63
2.5.3.1.8.3	ACPI Processor P-state Objects	63

2.5.3.1.8.4	Fixed ACPI Description Table (FADT) Entries	65
2.5.3.1.8.5	XPSS (Microsoft® Extended PSS) Object	65
2.5.3.2	Core C-states	65
2.5.3.2.1	C-state Names and Numbers	65
2.5.3.2.2	C-state Request Interface	65
2.5.3.2.3	C-state Actions	66
2.5.3.2.3.1	C-state Probes and Cache Flushing	66
2.5.3.2.3.2	Core C1 (CC1) State	66
2.5.3.2.3.3	Core C6 (CC6) State	66
2.5.3.2.3.4	Package C6 (PC6) State	67
2.5.3.2.4	C-state Request Monitors	67
2.5.3.2.4.1	FCH Messaging	67
2.5.3.2.4.2	Cache Flush On Halt Saturation Counter	67
2.5.3.2.5	Exiting C-states	68
2.5.3.2.6	ACPI Processor C-state Objects	68
2.5.3.2.6.1	_CST	68
2.5.3.2.6.2	_CRS	68
2.5.3.2.6.3	Fixed ACPI Description Table (FADT) Entries	68
2.5.3.2.7	BIOS Requirements for Initialization	69
2.5.3.3	Effective Frequency	69
2.5.4	NB Power Management	69
2.5.4.1	NB P-states	69
2.5.4.1.1	NB P-state Transitions	70
2.5.4.1.2	BIOS NB P-state Configuration	70
2.5.4.1.2.1	NB P-state COF and VID Synchronization After Warm Reset	70
2.5.4.1.2.2	NB P-state Transitions	71
2.5.4.1.2.3	NB P-state Configuration for Runtime	71
2.5.4.2	NB C-states	71
2.5.5	Bandwidth Requirements	72
2.5.6	GPU and Root Complex Power Management	72
2.5.6.1	Dynamic Power Management (DPM)	72
2.5.6.1.1	Activity Monitors	72
2.5.6.1.2	SCLK DPM	73
2.5.6.1.3	LCLK DPM	73
2.5.6.2	GPU and Root Complex Power Gating	73
2.5.7	DRAM Power Management	74
2.5.7.1	Memory P-states	74
2.5.7.2	DRAM Self-Refresh	74
2.5.7.3	Stutter Mode	75
2.5.7.3.1	System BIOS Requirements for Stutter Mode Operation During POST	75
2.5.7.4	EVENT_L	75
2.5.8	System Power Management	75
2.5.8.1	S-states	75
2.5.8.1.1	ACPI Suspend to RAM State (S3)	76
2.5.9	Application Power Management (APM)	76
2.5.9.1	Core Performance Boost (CPB)	76
2.5.9.1.1	C-state Boost	77
2.5.9.2	TDP Limiting	77
2.5.9.3	Bidirectional Application Power Management (BAPM)	77
2.6	Performance Monitoring	78
2.6.1	Performance Monitor Counters	78

2.6.1.1	Core Performance Monitor Counters	78
2.6.1.2	L2I Performance Monitor Counters	78
2.6.1.3	NB Performance Monitor Counters	79
2.6.2	Instruction Based Sampling (IBS)	79
2.7	Configuration Space	80
2.7.1	MMIO Configuration Coding Requirements	80
2.7.2	MMIO Configuration Ordering	81
2.7.3	Processor Configuration Space	81
2.8	Northbridge (NB)	81
2.8.1	NB Architecture	81
2.8.2	NB Routing	82
2.8.2.1	Address Space Routing	82
2.8.2.1.1	DRAM and MMIO Memory Space	82
2.8.2.1.2	IO Space	82
2.8.2.1.3	Configuration Space	82
2.8.2.1.3.1	Recommended Buffer Count Settings Overview	83
2.8.3	Memory Scrubbers	83
2.9	DRAM Controllers (DCTs)	84
2.9.1	Common DCT Definitions	84
2.9.2	DCT Frequency Support	85
2.9.3	DCT Configuration Registers	88
2.9.4	DDR Pad to Processor Pin Mapping	89
2.9.4.1	DDR Chip to Pad Mapping	89
2.9.5	DRAM Controller Direct Response Mode	90
2.9.6	DRAM Data Burst Mapping	90
2.9.7	DCT/DRAM Initialization and Resume	92
2.9.7.1	Low Voltage DDR3	92
2.9.7.2	NB P-state Specific Configuration	93
2.9.7.3	Memory P-state Specific Configuration	93
2.9.7.4	DDR Phy Initialization	94
2.9.7.4.1	Phy Voltage Level Programming	94
2.9.7.4.2	DRAM Channel Frequency Change	94
2.9.7.4.2.1	Requirements for DRAM Frequency Change During Training	95
2.9.7.4.3	Phy Fence Programming	95
2.9.7.4.3.1	Phy Fence Training	96
2.9.7.4.4	Phy Compensation Initialization	96
2.9.7.5	SPD ROM-Based Configuration	100
2.9.7.6	Non-SPD ROM-Based Configuration	101
2.9.7.6.1	TrdrdSdSc, TrdrdSdDc, and TrdrdDd (Read to Read Timing)	102
2.9.7.6.2	TwrwrSdSc, TwrwrSdDc, TwrwrDd (Write to Write Timing)	102
2.9.7.6.3	Twrrd (Write to Read DIMM Termination Turn-around)	102
2.9.7.6.4	TrwtTO (Read-to-Write Turnaround for Data, DQS Contention)	103
2.9.7.6.5	DRAM ODT Control	103
2.9.7.6.5.1	DRAM ODT Configuration	104
2.9.7.6.6	DRAM Address Timing and Output Driver Compensation Control	106
2.9.7.7	DCT Training Specific Configuration	114
2.9.7.8	DRAM Device and Controller Initialization	114
2.9.7.8.1	Software DDR3 Device Initialization	115
2.9.7.8.1.1	DDR3 MR Initialization	115
2.9.7.9	DRAM Training	118
2.9.7.9.1	Write Levelization Training	118

2.9.7.9.1.1	Write Leveling Seed Value	119
2.9.7.9.2	DQS Receiver Enable Training	120
2.9.7.9.2.1	DQS Receiver Enable Training Seed Value	121
2.9.7.9.3	DQS Receiver Enable Cycle Training	122
2.9.7.9.4	DQS Position Training	122
2.9.7.9.5	Calculating MaxRdLatency	126
2.9.7.9.5.1	MaxRdLatency Training	126
2.9.7.10	DRAM Phy Power Savings	127
2.9.8	Continuous Pattern Generation	128
2.9.8.1	DCT Training Pattern Generation	128
2.9.8.1.1	Activate and Precharge Command Generation	128
2.9.8.1.2	Read and Write Command Generation	129
2.9.8.1.3	Configurable Data Patterns	129
2.9.8.1.3.1	Static Data Pattern Override	131
2.9.8.1.3.2	Xor Data Pattern Override	132
2.9.8.1.4	Data Comparison	133
2.9.8.1.4.1	Activate and Precharge Traffic Generation	133
2.9.8.1.5	BubbleCnt and CmdStreamLen Programming	134
2.9.9	Memory Interleaving Modes	135
2.9.9.1	Chip Select Interleaving	135
2.9.10	Memory Hoisting	136
2.9.10.1	DramHoleOffset Programming	137
2.9.11	DRAM CC6/PC6 Storage	138
2.9.11.1	Fixed Storage	138
2.9.12	DRAM On DIMM Thermal Management and Power Capping	139
2.10	Thermal Functions	141
2.10.1	The Tctl Temperature Scale	141
2.10.2	Temperature Slew Rate Control	142
2.10.3	Temperature-Driven Logic	142
2.10.3.1	PROCHOT_L and Hardware Thermal Control (HTC)	142
2.10.3.2	Local Hardware Thermal Control (LHTC)	143
2.10.3.3	Software P-state Limit Control	143
2.10.3.4	THERMTRIP	143
2.11	Root Complex	144
2.11.1	Overview	144
2.11.2	Interrupt Routing	144
2.11.2.1	IOAPIC Configuration	145
2.11.3	Links	145
2.11.3.1	Overview	145
2.11.3.2	Link Configurations	146
2.11.4	Root Complex Configuration	146
2.11.4.1	LPC MMIO Requirements	146
2.11.4.2	Configuration for non-FCH Bridges	146
2.11.4.3	Link Configuration and Initialization	146
2.11.4.3.1	Link Configuration and Core Initialization	147
2.11.4.3.2	Link Training	147
2.11.4.4	Miscellaneous Features	147
2.11.4.4.1	Lane Reversal	147
2.11.4.4.2	Link Speed Changes	147
2.11.4.4.2.1	Autonomous Link Speed Changes	148
2.11.4.4.3	Deemphasis	148

2.11.4.5	Link Power Management	148
2.11.4.5.1	Link States	148
2.11.4.5.2	Dynamic Link-width Control	148
2.11.4.6	Link Test and Debug Features	148
2.11.4.6.1	Compliance Mode	148
2.11.5	BIOS Timer	149
2.11.6	PCIe® Client Interface Control	149
2.12	System Management Unit (SMU)	150
2.12.1	Software Interrupts	150
2.13	Graphics Processor (GPU)	150
2.13.1	Graphics Memory Controller (GMC)	150
2.13.2	Frame Buffer (FB)	151
2.14	RAS Features	152
2.14.1	Machine Check Architecture	152
2.14.1.1	Machine Check Registers	152
2.14.1.2	Machine Check Errors	153
2.14.1.3	Error Detection, Action, Logging, and Reporting	154
2.14.1.3.1	MCA conditions that cause Shutdown	155
2.14.1.3.2	Error Logging During Overflow	155
2.14.1.4	MCA Initialization	156
2.14.1.5	Error Code	156
2.14.1.6	Handling Machine Check Exceptions	158
2.14.1.6.1	Differentiation Between System-Fatal and Process-Fatal Errors	160
2.14.1.7	Error Thresholding	161
2.14.1.8	Scrub Rate Considerations	162
2.14.1.9	Error Diagnosis	163
2.14.1.9.1	Common Diagnosis Information	164
2.14.2	DRAM ECC Considerations	165
2.14.2.1	ECC Syndromes	165
2.14.2.1.1	x4 ECC	165
2.14.3	Error Injection and Simulation	167
2.14.3.1	DRAM Error Injection	167
2.15	Fusion Controller Hub	169
2.15.1	MMIO Programming for Legacy Devices	169
2.15.2	USB Controllers	169
2.15.2.1	USB Power Management	170
2.15.2.2	USB Interrupts	170
2.15.2.3	BIOS Programming Requirements For USB Reset During S3 Resume	170
2.15.2.4	Enabling the xHCI Controller	170
2.15.2.5	OHCI Arbiter Mode	170
2.15.2.6	USB2.0 Controller PHY Configuration and Calibration	171
2.15.2.7	USB2.0 Controller Common PHY Impedance Calibration	171
2.15.2.8	USB2.0 Controller ISO Device CRC False Error Detection	171
2.15.2.9	xHC USB2.0 Common PHY Calibration	172
2.15.2.10	USB3.0 PHY Auto-Calibration Enablement	173
2.15.2.11	xHCI ISO Device CRC False Error Detection	173
2.15.2.12	xHCI PHY Clock Gating	173
2.15.2.13	xHCI Firmware Preload	173
2.15.2.14	xHCI Clear Pending PME on Sx State Entry	174
2.15.2.15	xHCI Enable OHCI3/EHCI3 on S4/S5 State Entry	174
2.15.3	SATA	174



2.15.3.1	SATA Operating Mode	174
2.15.3.2	SATA Drive Detection in IDE mode	174
2.15.3.3	SATA PHY Auto-Calibration Enablement	174
2.15.3.4	SATA PHY Fine Tuning	175
2.15.3.5	SATA PHY Reference Clock Selection	176
2.15.3.6	SATA Power Management	176
2.15.3.6.1	SATA PHY Power Saving	176
2.15.3.7	Enable Shadow Register Reload	176
2.15.3.8	SATA Interrupt Handling	177
2.15.3.8.1	Line Interrupt	177
2.15.3.8.2	MSI Message	177
2.15.3.9	Clear status of SATA PERR	177
2.15.4	LPC Bus Interface	177
2.15.4.1	Enabling LPC DMA function	177
2.15.4.2	Enabling SPI 100	177
2.15.5	SD Controller	177
2.15.6	HD Audio	178
2.15.6.1	HD Audio AF and MSI Capability	178
2.15.7	ASF Controller	178
2.15.8	Integrated Micro-Controller	179
2.15.9	On-Chip Clock Generator	179
2.15.9.1	Power Saving In Internal Clock Mode	179
2.15.9.2	Global A-Link / B-Link Clock Gating	179
2.15.9.3	CG_PLL CMOS Clock Driver Setting for Power Saving	179
2.15.10	FCH Gasket	180
2.15.11	A-Link Bridge	180
2.15.11.1	Detection of Upstream Interrupts	180
2.15.11.2	AB Memory Power Saving	180
2.15.11.3	AB Internal Clock Gating	180
2.15.11.4	AB 32/64 Byte DMA Write Enable	180
<b>3</b>	<b>Registers</b>	<b>181</b>
3.1	Register Descriptions and Mnemonics	181
3.1.1	Northbridge MSRs In Multi-Core Products	185
3.1.2	Software Recommendation (BIOS, SBIOS)	185
3.1.3	See Keyword (See:)	185
3.1.4	Mapping Tables	186
3.1.4.1	Register Mapping	186
3.1.4.2	Index Mapping	186
3.1.4.3	Field Mapping	186
3.1.4.4	Broadcast Mapping	186
3.1.4.5	Reset Mapping	186
3.1.4.6	Valid Values	186
3.1.4.7	BIOS Recommendations	187
3.2	IO Space Registers	187
3.3	Device 0 Function 0 (Root Complex) Configuration Registers	189
3.4	Device 0 Function 2 Configuration Registers	229
3.5	Device 1 Function 0 (Internal Graphics) Configuration Registers	231
3.6	Device 1 Function 1 (Audio Controller) Configuration Registers	243
3.7	Device 2 Function 0 (Host Bridge) Configuration Registers	254
3.8	Device 2 Function [5:1] (Root Port) Configuration Registers	255

3.9	Device 18h Function 0 Configuration Registers	286
3.10	Device 18h Function 1 Configuration Registers	299
3.11	Device 18h Function 2 Configuration Registers	311
3.12	Device 18h Function 3 Configuration Registers	395
3.13	Device 18h Function 4 Configuration Registers	431
3.14	Device 18h Function 5 Configuration Registers	440
3.15	Northbridge IOAPIC Registers	455
3.16	APIC Registers	457
3.17	CPUID Instruction Registers	467
3.18	MSRs - MSR0000_xxxx	495
3.19	MSRs - MSRC000_0xxx	527
3.20	MSRs - MSRC001_0xxx	533
3.21	MSRs - MSRC001_1xxx	561
3.22	Core Performance Counter Events	576
3.22.1	PMCx0[1F:00] Events (FP)	576
3.22.2	PMCx0[3F:20] Events (LS)	577
3.22.3	PMCx0[5F:40] Events (DC)	579
3.22.4	PMCx[8,1:0][7F:60] Events (BU)	581
3.22.5	PMCx[1:0][9F:80] Events (IC)	585
3.22.6	PMCx[1,0][DF:C0] Events (EX, DE)	587
3.23	L2I Performance Counter Events	590
3.24	NB Performance Counter Events	592
3.24.1	PMCx0E[7:4] Events (Memory Controller)	592
3.24.2	PMCx0E[F:8] Events (Crossbar)	592
3.24.3	PMCx0F[F:0] Events (Crossbar)	595
3.24.4	NBPMCx1E[F:0] Events (Crossbar)	595
3.24.5	NBPMCx1F[F:0] Events (Memory Controller, Crossbar)	598
3.25	Fusion Controller Hub Registers	601
3.25.1	Legacy Block Configuration Registers (IO)	601
3.25.2	AB Configuration Registers	621
3.25.3	SATA Controller	624
3.25.3.1	Device 11h Function 0 (SATA) Configuration Registers	624
3.25.3.2	SATA IO Registers	640
3.25.3.2.1	IDE Compatibility Mode and Native Mode Registers	640
3.25.3.2.2	IDE Bus Master Registers	642
3.25.3.3	SATA Memory Mapped AHCI Registers	643
3.25.3.3.1	Generic Host Control Registers	644
3.25.3.3.2	Port Control Registers	650
3.25.3.3.3	Enclosure Buffer Management Registers	663
3.25.4	USB Controllers	666
3.25.4.1	USB 1.1 (OHCI)	666
3.25.4.1.1	Devices 16h, 13h, 12h Function 0 (OHCI) Configuration Registers	666
3.25.4.1.2	OHCI Memory Mapped IO Registers	674
3.25.4.2	USB 2.0 (EHCI)	685
3.25.4.2.1	Devices 16h, 13h, 12h Function 2 (EHCI) Configuration Registers	686
3.25.4.2.2	EHCI Memory Mapped IO Registers	695
3.25.4.3	USB 3.0 (xHCI)	709
3.25.4.3.1	Device 10h Function 0 (xHCI) Configuration Registers	709
3.25.4.3.1.1	USB xHCI Capability Registers	727
3.25.4.3.2	xHCI Power Management Registers	728
3.25.5	HD Audio Controller	734

3.25.5.1	Device 14h Function 2 (Audio Controller) Configuration Registers . . .	734
3.25.6	Secure Digital (SD) Controller . . . . .	741
3.25.6.1	Device 14h Function 7 Configuration Registers (SD) . . . . .	741
3.25.6.2	SD Host Controller Configuration Registers (SDHC) . . . . .	746
3.25.7	SMBus Host Controller . . . . .	762
3.25.7.1	Device 14h Function 0 (SMBus) Configuration Registers . . . . .	762
3.25.7.2	ASF (Alert Standard Format) Registers . . . . .	765
3.25.7.3	SMBus Registers . . . . .	770
3.25.8	IOAPIC Registers . . . . .	777
3.25.9	LPC-ISA Bridge. . . . .	779
3.25.9.1	Device 14h Function 3 (LPC Bridge) Configuration Registers . . . . .	779
3.25.9.2	SPI Registers . . . . .	792
3.25.10	High Precision Event Timer (HPET) Registers. . . . .	801
3.25.11	Miscellaneous (MISC) Registers. . . . .	804
3.25.12	GPIO Pin control registers. . . . .	827
3.25.12.1	GPIO Registers . . . . .	827
3.25.12.2	IOMux Registers . . . . .	828
3.25.13	Power Management (PM) Registers . . . . .	829
3.25.14	Power Management Block 2 (PM2) Registers. . . . .	861
3.25.15	Standard ACPI Registers . . . . .	871
3.25.15.1	AcpiPmEvtBlk . . . . .	871
3.25.15.2	AcpiPm1CntBlk . . . . .	872
3.25.15.3	AcpiPm2CntBlk . . . . .	873
3.25.15.4	AcpiPmTmrBlk . . . . .	873
3.25.15.5	CpuCntBlk . . . . .	873
3.25.15.6	AcpiGpe0Blk . . . . .	874
3.25.15.7	SmiCmdBlk . . . . .	874
3.25.16	SMI Registers. . . . .	875
3.25.17	Watchdog Timer (WDT) Registers . . . . .	903
3.25.18	Wake Alarm Device (AcDcTimer) Registers . . . . .	903
<b>4</b>	<b>Register List. . . . .</b>	<b>907</b>

# List of Figures

Figure 1:	A L2 Complex.....	29
Figure 2:	System Diagram.....	30
Figure 3:	DQS Position Training Example Results.....	124
Figure 4:	DQS Position Training Insertion Delay Recovery Example Results.....	125
Figure 5:	Memory Configuration with Memory Hole inside of Region.....	137
Figure 6:	Memory Configuration with Memory Hole outside of Region.....	138
Figure 7:	Tctl scale.....	141
Figure 8:	Root complex topology.....	144
Figure 9:	Address/Command Timing at the Processor Pins.....	326

# List of Tables

Table 1:	Arithmetic and Logical Operators.....	21
Table 2:	Functions.....	21
Table 3:	Operator Precedence and Associativity .....	22
Table 4:	Definitions.....	22
Table 5:	Processor revision conventions.....	28
Table 6:	SMM Initial State.....	42
Table 7:	SMM Save State.....	43
Table 8:	Power Management Support.....	52
Table 9:	Software P-state Naming .....	58
Table 10:	Software P-state Control .....	59
Table 11:	ONION Link Definitions .....	83
Table 12:	DCT Definitions.....	84
Table 13:	DDR3 UDIMM Maximum Frequency Support with 6-layer Motherboard Design FT3 Package .....	85
Table 14:	DDR3 UDIMM Maximum Frequency Support with 6-layer Motherboard Design FT3 Package Microserver.....	86
Table 15:	DDR3 SODIMM Maximum Frequency Support with 6-layer Motherboard Design FT3, FS1b Package.....	86
Table 16:	DDR3 UDIMM Maximum Frequency Support with 4-layer Motherboard Design FT3 Package .....	86
Table 17:	DDR3 SODIMM Maximum Frequency Support with 4-layer Motherboard Design FT3, FS1b Package .....	87
Table 18:	DDR3 SODIMM plus Solder-down DRAM Maximum Frequency with 6-layer Motherboard FT3 Package.....	87
Table 19:	DDR3 SODIMM plus Solder-down DRAM Maximum Frequency with 4-layer Motherboard FT3 package.....	87
Table 20:	DDR3 Solder-down DRAM Maximum Frequency Support FT3 Package.....	88
Table 21:	DDR3 UDIMM Maximum Frequency Support FS1b Package .....	88
Table 22:	DDR3 Population Support .....	88
Table 23:	Package pin mapping .....	89
Table 24:	Pad (from chiplet) pin mapping .....	90
Table 25:	DDR PLL Lock Time.....	95
Table 26:	Phy predriver calibration codes for Data/DQS at 1.5V .....	97
Table 27:	Phy predriver calibration codes for Data/DQS at 1.35V .....	97
Table 28:	Phy Predriver Calibration Codes for Data/DQS at 1.25V .....	98
Table 29:	Phy predriver calibration codes for Cmd/Addr at 1.5V .....	98
Table 30:	Phy predriver calibration codes for Cmd/Addr at 1.35V .....	99
Table 31:	Phy Predriver Calibration Codes for Cmd/Addr at 1.25V .....	99
Table 32:	Phy predriver calibration codes for Clock at 1.5V.....	99
Table 33:	Phy predriver calibration codes for Clock at 1.35V.....	100
Table 34:	Phy Predriver Calibration Codes for Clock at 1.25V.....	100
Table 35:	DDR3 ODT Pattern NumDimmSlots=1 .....	103
Table 36:	DDR3 ODT Pattern NumDimmSlots=2 .....	103

Table 37:	BIOS Recommendations for MR1[RttNom] and MR2[RttWr] UDIMM FT3 Package .....	104
Table 38:	BIOS Recommendations for MR1[RttNom] and MR2[RttWr] SODIMM FT3, FS1b Package .....	104
Table 39:	BIOS Recommendations for MR1[RttNom] and MR2[RttWr] SODIMM plus Solder-down DRAM FT3 Package.....	105
Table 40:	BIOS Recommendations for MR1[RttNom] and MR2[RttWr] Solder-down DRAM FT3 Package.....	105
Table 41:	BIOS Recommendations for MR1[RttNom] and MR2[RttWr] UDIMM FS1b Package .....	105
Table 42:	BIOS Recommendations for SlowAccessMode, POdtOff, Addr/Cmd Timings and Output Driver Control UDIMM FT3 Package.....	107
Table 43:	BIOS Recommendations for SlowAccessMode, POdtOff, Addr/Cmd Timings and Output Driver Control SODIMM FT3, FS1b Package.....	109
Table 44:	BIOS Recommendations for SlowAccessMode, POdtOff, Addr/Cmd Timings and Output Driver Control SODIMM plus Solder-down DRAM FT3 Package.....	111
Table 45:	BIOS Recommendations for SlowAccessMode, POdtOff, Addr/Cmd Timings and Output Driver Control Solder-down DRAM FT3 Package.....	112
Table 46:	BIOS Recommendations for SlowAccessMode, POdtOff, Addr/Cmd Timings and Output Driver Control UDIMM FS1b Package.....	113
Table 47:	DCT Training Specific Register Values .....	114
Table 48:	BIOS Recommendations for MR0[WR].....	115
Table 49:	BIOS Recommendations for MR0[CL[3:0]] .....	116
Table 50:	BIOS Recommendations for MR2[ASR, SRT] .....	117
Table 51:	DDR3 Write Leveling Seed Values.....	120
Table 52:	DDR3 DQS Receiver Enable Training Seed Values.....	122
Table 53:	Configurable Data Pattern Example with 1 Address Target .....	130
Table 54:	Configurable Data Pattern Circular Shift Example with 1 Address Target .....	131
Table 55:	Data Pattern Override Example with 1 Address Target .....	132
Table 56:	Command Generation and Data Comparison .....	133
Table 57:	DDR3 Command Generation and BubbleCnt Programming.....	134
Table 58:	Recommended Interleave Configurations.....	135
Table 59:	DDR3 Swapped Normalized Address Lines for CS Interleaving.....	136
Table 60:	Example storage region configuration .....	139
Table 61:	Recommended Interrupt Routing and Swizzling Configuration.....	145
Table 62:	Supported General Purpose (GPP) Link Configurations .....	146
Table 63:	SMU Software Interrupts .....	150
Table 64:	Recommended Frame Buffer Configurations .....	151
Table 65:	MCA register cross-reference table .....	153
Table 66:	Overwrite Priorities for All Banks .....	156
Table 67:	Error Code Types .....	157
Table 68:	Error codes: transaction type (TT).....	157
Table 69:	Error codes: cache level (LL).....	157
Table 70:	Error codes: memory transaction type (RRRR).....	157
Table 71:	Error codes: participation processor (PP) .....	158
Table 72:	Error codes: memory or IO (II).....	158
Table 73:	Error codes: Internal Error Type (UU).....	158
Table 74:	Error Scope Hierarchy .....	160

Table 75:	Recommended Scrub Rates per Node.....	162
Table 76:	Registers Commonly Used for Diagnosis.....	164
Table 77:	x4 ECC Correctable Syndromes .....	166
Table 78:	USB Port Mapping.....	169
Table 79:	USB Port to EHCI[3:1]xB4[PortNumber] Mapping .....	172
Table 80:	USB Port to D10F0x4C_x4000_0000[PortNumber] Mapping .....	173
Table 81:	SATA PHY Power Saving States .....	176
Table 82:	HD Audio AF and MSI Capability Settings .....	178
Table 83:	ASF Remote Control Commands .....	179
Table 84:	Terminology in Register Descriptions .....	183
Table 85:	Reset values for D0F0x64_x3[4:0].....	194
Table 86:	Register Mapping for D0F0xBC_x3FD[8C:00:step14].....	204
Table 87:	Register Mapping for D0F0xBC_x3FD[94:08:step14] .....	205
Table 88:	Register Mapping for D0F0xBC_x3FD[9C:10:step14].....	205
Table 89:	Index addresses for D0F0xE4_x0110_001[8:7,3:2] .....	212
Table 90:	Index address mapping for D0F0xE4_x0130_0[C:8]00.....	215
Table 91:	Index address mapping for D0F0xE4_x0130_0[C:8]03.....	216
Table 92:	Index address mapping for D0F0xE4_x0130_0[C:8]05.....	216
Table 93:	Lane index addresses for D0F0xE4_x0130_802[4:1] .....	220
Table 94:	Reset Mapping for D0F0xE4_x0130_802[4:1] .....	220
Table 95:	Field mapping for D0F0xE4_x0130_802[4:1] .....	220
Table 96:	Lane index addresses for D0F0xE4_x0130_802[8:5] .....	221
Table 97:	Reset Mapping for D0F0xE4_x0130_802[8:5] .....	221
Table 98:	Field mapping for D0F0xE4_x0130_802[8:5] .....	221
Table 99:	Register Mapping for D2F[5:1]x00 .....	255
Table 100:	Link controller state encodings.....	277
Table 101:	Register Mapping for D18F0x[5C:40].....	287
Table 102:	Register Mapping for D18F0x[E4,C4,A4,84] .....	291
Table 103:	Register Mapping for D18F0x[EC,CC,AC,8C].....	292
Table 104:	Register Mapping for D18F0x[F0,D0,B0,90].....	292
Table 105:	Link Buffer Definitions.....	293
Table 106:	Register Mapping for D18F0x[F4,D4,B4,94].....	294
Table 107:	Register Mapping for D18F0x[F8,D8,B8,98].....	295
Table 108:	Register Mapping for D18F0x[11C,118,114,110].....	296
Table 109:	Register Mapping for D18F0x[18C:170].....	297
Table 110:	Onion Definitions.....	297
Table 111:	Register Mapping for D18F1x[17C:140,7C:40].....	299
Table 112:	Register Mapping for D18F1x[7:4][8,0].....	300
Table 113:	Register Mapping for D18F1x[7:4][C,4] .....	300
Table 114:	Register Mapping for D18F1x[2CC:2A0,1CC:180,BC:80] .....	301
Table 115:	Register Mapping for D18F1x[2B:1A,B:8][8,0] .....	302
Table 116:	Register Mapping for D18F1x[2B:1A,B:8][C,4].....	302
Table 117:	Register Mapping for D18F1x[1F:1E,D:C][8,0] .....	304
Table 118:	Register Mapping for D18F1x[1F:1E,D:C][C,4].....	305

Table 119:	Register Mapping for D18F1x[1DC:1D0,EC:E0] .....	305
Table 120:	Register Mapping for D18F1x2[1C:00].....	309
Table 121:	Register Mapping for D18F1x2[1,0][8,0].....	309
Table 122:	Register Mapping for D18F1x2[1,0][C,4] .....	310
Table 123:	DIMM, Chip Select, and Register Mapping .....	311
Table 124:	DDR3 DRAM Address Mapping.....	315
Table 125:	Valid Values for Memory Clock Frequency Value Definition .....	321
Table 126:	Index Mapping for D18F2x9C_x0000_0[3:0]0[3:1]_dct[0]_mp[1:0].....	325
Table 127:	Byte Lane Mapping for D18F2x9C_x0000_0[3:0]0[3:1]_dct[0]_mp[1:0] .....	325
Table 128:	Index Mapping for D18F2x9C_x0000_0[3:0]0[7:5]_dct[0]_mp[1:0].....	327
Table 129:	Byte Lane Mapping for D18F2x9C_x0000_0[3:0]0[7:5]_dct[0]_mp[1:0] .....	327
Table 130:	Index Mapping for D18F2x9C_x0000_00[2A:10]_dct[0]_mp[1:0].....	331
Table 131:	Byte Lane Mapping for D18F2x9C_x0000_00[2A:10]_dct[0]_mp[1:0] .....	332
Table 132:	Index Mapping for D18F2x9C_x0000_00[4A:30]_dct[0]_mp[1:0].....	333
Table 133:	Byte Lane Mapping for D18F2x9C_x0000_00[4A:30]_dct[0]_mp[1:0] .....	334
Table 134:	Index Mapping for D18F2x9C_x0000_00[52:50]_dct[0] .....	335
Table 135:	Byte Lane Mapping for D18F2x9C_x0000_00[52:50]_dct[0] .....	335
Table 136:	Index Mapping for D18F2x9C_x0D0F_0[F,8:0]02_dct[0].....	336
Table 137:	Broadcast Mapping for D18F2x9C_x0D0F_0[F,8:0]02_dct[0] .....	336
Table 138:	Valid Values for D18F2x9C_x0D0F_0[F,8:0]02_dct[0][TxPreP].....	336
Table 139:	Index Mapping for D18F2x9C_x0D0F_0[F,8:0]0[B,7,3]_dct[0].....	337
Table 140:	Index Mapping for D18F2x9C_x0D0F_0[F,8:0]04_dct[0]_mp[1:0] .....	337
Table 141:	Index Mapping for D18F2x9C_x0D0F_0[F,8:0]06_dct[0] .....	338
Table 142:	Index Mapping for D18F2x9C_x0D0F_0[F,8:0]0A_dct[0] .....	338
Table 143:	Broadcast Mapping for D18F2x9C_x0D0F_0[F,8:0]0[A,6]_dct[0].....	338
Table 144:	Index Mapping for D18F2x9C_x0D0F_0[F,8:0]10_dct[0]_mp[1:0] .....	339
Table 145:	Index Mapping for D18F2x9C_x0D0F_0[F,8:0]13_dct[0]_mp[1:0] .....	339
Table 146:	Index Mapping for D18F2x9C_x0D0F_0[F,8:0]1C_dct[0]_mp[1:0] .....	340
Table 147:	Index Mapping for D18F2x9C_x0D0F_0[F,8:0]1E_dct[0]_mp[1:0].....	341
Table 148:	Index Mapping for D18F2x9C_x0D0F_0[F,8:0]1F_dct[0]_mp[1:0].....	342
Table 149:	Index Mapping for D18F2x9C_x0D0F_0[F,8:0]2[3:0]_dct[0]_mp[1:0] .....	342
Table 150:	Index Addresses for D18F2x9C_x0D0F_0[F,8:0]30_dct[0] .....	344
Table 151:	Broadcast Write Index Address for D18F2x9C_x0D0F_0[F,8:0]30_dct[0] .....	344
Table 152:	Index addresses for D18F2x9C_x0D0F_0[F,8:0]31_dct[0] .....	345
Table 153:	Broadcast write index address for D18F2x9C_x0D0F_0[F,8:0]31_dct[0] .....	345
Table 154:	Index addresses for D18F2x9C_x0D0F_0[F,8:0]38_dct[0] .....	345
Table 155:	Broadcast write index address for D18F2x9C_x0D0F_0[F,8:0]38_dct[0] .....	346
Table 156:	Broadcast write index address for D18F2x9C_x0D0F_1C00_dct[0].....	346
Table 157:	Index Mapping for D18F2x9C_x0D0F_2[2:0]02_dct[0] .....	346
Table 158:	Index address mapping for D18F2x9C_x0D0F_[C,8,2][2:0]1E_dct[0]_mp[1:0].....	347
Table 159:	Index address mapping for D18F2x9C_x0D0F_[C,8,2][2:0]1F_dct[0] .....	347
Table 160:	Index Addresses for D18F2x9C_x0D0F_[C,8,2][2:0]20_dct[0]_mp[1:0].....	348
Table 161:	Index Addresses for D18F2x9C_x0D0F_2[F,2:0]30_dct[0] .....	349
Table 162:	Index Mapping for D18F2x9C_x0D0F_[C,8][1:0]02_dct[0].....	350



Table 163:	Index Mapping for D18F2x9C_x0D0F_[C,8][1:0][12,0E,0A,06]_dct[0].....	351
Table 164:	Index Mapping for D18F2x9C_x0D0F_[C,8][F:0]30_dct[0].....	352
Table 165:	BIOS Recommendations for D18F2x1B[4:0].....	365
Table 166:	Field Mapping for D18F2x1BC_dct[0] .....	368
Table 167:	BIOS Recommendations for D18F2x1BC_dct[0] .....	368
Table 168:	BIOS Recommendations for RdPtrInit .....	372
Table 169:	Register Mapping for D18F2x25[8,4]_dct[0].....	382
Table 170:	Register Mapping for D18F2x2[B4,B0,AC,A8]_dct[0] .....	389
Table 171:	Field Mapping for D18F2x2[B4,B0,AC,A8]_dct[0] .....	389
Table 172:	Field Mappings for D18F2x2[C0,BC]_dct[0] .....	390
Table 173:	Buffer Definitions .....	405
Table 174:	SMAF Action Definition .....	409
Table 175:	Register Mapping for D18F3x1[54:48] .....	423
Table 176:	D18F5x80[Enabled, DualCore, TripleCore, QuadCore] Definition.....	441
Table 177:	Register Mapping for D18F5x16[C:0].....	446
Table 178:	NB P-state Definitions .....	446
Table 179:	Register Mapping for APIC[170:100] .....	459
Table 180:	Register Mapping for APIC[1F0:180] .....	460
Table 181:	Register Mapping for APIC[270:200] .....	460
Table 182:	ICR valid combinations .....	461
Table 183:	Register Mapping for APIC3[60:50] .....	463
Table 184:	Div[3,1:0] Value Table.....	464
Table 185:	Register Mapping for APIC[4F0:480] .....	466
Table 186:	Register Mapping for APIC[530:500] .....	466
Table 187:	Reset Mapping for CPUID Fn8000_0000_E[D,C,B]X .....	467
Table 188:	CPUID Fn8000_0000_E[B,C,D]X Value .....	475
Table 189:	Valid Values for CPUID Fn8000_000[4:2]_E[D,C,B,A]X.....	478
Table 190:	ECX mapping to Cache Type for CPUID Fn8000_001D_E[D,C,B,A]X.....	489
Table 191:	Register Mapping for MSR0000_020[E,C,A,8,6,4,2,0] .....	499
Table 192:	Valid Values for Memory Type Definition.....	500
Table 193:	Register Mapping for MSR0000_020[F,D,B,9,7,5,3,1] .....	500
Table 194:	Register Mapping for MSR0000_02[6F:68,59:58,50].....	501
Table 195:	Field Mapping for MSR0000_02[6F:68,59:58,50].....	501
Table 196:	MC0 Error Descriptions.....	505
Table 197:	MC0 Error Signatures .....	506
Table 198:	MC0 Address Register.....	507
Table 199:	MC1 Error Descriptions.....	509
Table 200:	MC1 Error Signatures .....	509
Table 201:	MC1 Address Register.....	510
Table 202:	MC2 Error Descriptions.....	512
Table 203:	MC2 Error Signatures .....	513
Table 204:	MC2 Address Register.....	514
Table 205:	MC4 Error Descriptions.....	517
Table 206:	MC4 Error Signatures, Part 1.....	519

Table 207:	MC4 Error Signatures, Part 2.....	519
Table 208:	Format of MSR0000_0412[ErrAddr[47:1]] for All Other Errors.....	520
Table 209:	Format of MSR0000_0412[ErrAddr[47:1]] for Protocol Errors .....	520
Table 210:	Valid Values for ProtocolErrorType .....	521
Table 211:	Format of MSR0000_0412[ErrAddr[47:1]] for NB Array Errors .....	521
Table 212:	Valid Values for ArrayErrorType .....	521
Table 213:	Format of MSR0000_0412[ErrAddr[47:1]] for Watchdog Timer Errors .....	522
Table 214:	MC5 Error Descriptions.....	525
Table 215:	MC5 Error Signatures .....	525
Table 216:	MC5 Address Register.....	526
Table 217:	Register Mapping for MSRC001_00[03:00] .....	533
Table 218:	Register Mapping for MSRC001_00[07:04] .....	534
Table 219:	Register Mapping for MSRC001_00[35:30] .....	540
Table 220:	BIOS Recommendations for MSRC001_00[35:30] .....	541
Table 221:	Register Mapping for MSRC001_00[53:50] .....	545
Table 222:	Register Mapping for MSRC001_00[6B:64].....	548
Table 223:	P-state Definitions.....	549
Table 224:	Register Mapping for MSRC001_023[6,4,2,0] .....	558
Table 225:	Register Mapping for MSRC001_023[7,5,3,1] .....	559
Table 226:	Register Mapping for MSRC001_024[6,4,2,0] .....	559
Table 227:	Register Mapping for MSRC001_024[7,5,3,1] .....	560
Table 228:	Register Mapping for MSRC001_101[B:9].....	564
Table 229:	Field Mapping for MSRC001_101[B:9].....	565
Table 230:	Register Mapping for PMCx0D[F:C] .....	589
Table 231:	SATA Controller Subclass Code and ProgramIF Settings .....	625
Table 232:	IDE Compatibility Mode and Native Mode Address Mapping .....	640
Table 233:	OHCI[3:1]x48 reset values .....	679
Table 234:	Reset mapping for EHCI[3:1]x04 .....	695
Table 235:	Register Mapping for SDHCx6[C:0:step4].....	758
Table 236:	Field Mapping for SDHCx6[C:0:step4].....	759
Table 237:	Reset Mapping for IOMUXx[E4:00] .....	828
Table 238:	BIOS Recommendations for KbRstEn .....	852
Table 239:	BIOS Recommendations for UsbPhyS5PwrDwnEnable.....	860

# Revision History

KB BKDG Revision 3.03 Changes, Feb 19, 2015, PUB release

- D0F0x64\_x1D[Vga16En]: Updated BIOS recommendation.
- D2F[5:1]xE4\_xA0[Lc16xClearTxPipe]: Updated BIOS recommendation.
- D10F0x10: Updated.
- D14F3x48[IOPortEnable4]: Updated Reset.
- D18F3x78[RspDBC]: Updated BIOS recommendation.
- PMxE4: Added.

KB BKDG Revision 3.02 Changes, Dec 16, 2013, PUB release

- D0F0xBC[NbSmuIndData]: Updated.
- D18F4x16C[TdpLimitDis]: Updated.
- IOC00 [Pci\_Intr\_Index]: Updated.
- MSRC001\_1002 [CPUID Features for CPUID Fn0000\_0007\_E[B,A]X\_x0]: Updated.

KB BKDG Revision 3.01 Changes, Oct 31, 2013, PUB release

- 2.5.1.2 [Internal VID Registers and Encodings]: Updated.
- 2.9.2 [DCT Frequency Support]: Updated.
- 2.9.7.4 [DDR Phy Initialization]: Updated.
- 2.9.7.6.5.1 [DRAM ODT Configuration]: Updated.
- 2.9.7.6.6 [DRAM Address Timing and Output Driver Compensation Control]: Updated.
- 2.10.3.1 [PROCHOT\_L and Hardware Thermal Control (HTC)]: Updated.
- 2.10.3.4 [THERMTRIP]: Updated.
- 2.15.2.7 [USB2.0 Controller Common PHY Impedance Calibration]: Updated.
- D0F0x64\_x1F [FCH Location]: Updated.
- D10F0x4C\_x4000\_0008[AddToCommonCalibration, NewCalBus, UseCommonCalibration]: Updated.
- D11F0x98 [PHY Fine Tune PortX GenX Setting]: Updated.
- D11F0x9C [PortX Setting 2]: Updated.
- D14F3xB8[LpcBusPullUpEn]: Updated.
- D14F7xB8 [SD Auto Pattern]: Updated.
- D18F2x94\_dct[0] [DRAM Configuration High]: Updated.
- D18F2x9C\_x0D0F\_E000\_dct[0]\_mp[1:0][POdtOff, 12:4]: Updated.
- D18F2x9C\_x0D0F\_E007\_dct[0] [Phy DLL Lock Time]: Updated.
- D18F3x180 [Extended NB MCA Configuration]: Updated.
- D18F3x1FC[USB2CompCal[3:0].Value, 31:5]: Updated.
- D18F5x178[ProcHotToGnbEn]: Updated.
- D18F5xE8[Tdp2Watt, 15:10]: Updated.
- EHCI[3:1]xC0[AddToCommonCalibration, NewCalBus, UseCommonCalibration]: Updated.
- IO073\_x0B[DaylightSavingEnable]: Updated.
- MISCx40 [MiscClkCntl1]: Updated.
- PMx2E [Smbus0Sel]: Updated.
- PMxC4[ResetButtonEn]: Updated.
- PMxCC [IoDrvSth]: Updated.
- PMxD2 [PmioDebug]: Updated.
- SMIX98[28]: Updated.
- SMIX9C[31:0]: Updated.
- XHCI\_PMx10[CcuMode]: Updated.

KB BKDG Revision 3.00 initial release.

## 1 Overview

This document defines AMD Family 16h Models 00h-0Fh Processors, henceforth referred to as the processor.

- The processor overview is located at [2.1 \[Processor Overview\]](#).
- The processor is distinguished by the combined ExtFamily and BaseFamily fields of the CPUID instruction (see [CPUID Fn8000\\_0001\\_EAX](#) in [3.17 \[CPUID Instruction Registers\]](#)).

### 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](http://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 and 256-Bit Media Instructions, #26568.
- AMD64 Architecture Programmer's Manual Volume 5: 64-Bit Media and x87 Floating-Point Instructions, #26569.
- Software Optimization Guide for AMD Family 16h Processors, #52128.
- Revision Guide for AMD Family 16h Models 00h-0Fh Processors, #51810.
- JEDEC standards. [www.jedec.org](http://www.jedec.org).
- PCI local bus specification. ([www.pcisig.org](http://www.pcisig.org)).
- PCI Express® specification. ([www.pcisig.org](http://www.pcisig.org)).
- Universal Serial Bus Specification (<http://www.usb.org>)
- Serial ATA Specification (<http://www.sata-io.org>)
- AT Attachment with Packet Interface (<http://www.t13.org>)
- SD Host Controller Standard Specification (<https://www.sdcard.org>)
- Alert Standard Format Specification (<http://dmtf.org/standards/asf>)

### 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 [3.1 \[Register Descriptions and Mnemonics\]](#); register mnemonics all utilize hexadecimal numbering.
- **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 generally follow Verilog conventions for logic equations.

**Table 1: Arithmetic and Logical Operators**

Operator	Definition
{}	Concatenation. 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 LSB's are Xlate[3:0].
	Bitwise OR. E.g. (01b   10b == 11b).
	Logical OR. E.g. (01b    10b == 1b); treats multibit operand as 1 if >=1 and produces a 1-bit result.
&	Bitwise AND. E.g. (01b & 10b == 00b).
&&	Logical AND. E.g. (01b && 10b == 1b); logical treats multibit operand as 1 if >=1 and produces a 1-bit result.
^	Bitwise exclusive-OR. E.g. (01b ^ 10b == 11b). Sometimes used as “raised to the power of” as well, as indicated by the context in which it is used. E.g. (2^2 == 4).
~	Bitwise NOT. (also known as one's complement). E.g. (~10b == 01b).
!	Logical NOT. E.g. (!10b == 0b); treats multibit operand as 1 if >=1 and produces a 1-bit result.
<, <=, >, >=, ==, !=	Relational. Less than, Less than or equal, greater, greater than or equal, equal, and not equal.
+, -, *, /, %	Arithmetic. Addition, subtraction, multiplication, division, and modulus.
<<	Bitwise left shift. Shift left first operand by the number of bits specified by the 2nd operand. E.g. (01b << 01b == 10b).
>>	Bitwise right shift. Shift right first operand by the number of bits specified by the 2nd operand. E.g. (10b >> 01b == 01b).
?:	Ternary conditional. E.g. <i>condition ? value if true : value if false</i> . Equivalent to IF <i>condition</i> THEN <i>value if true</i> ELSE <i>value if false</i> .

**Table 2: Functions**

Function	Definition
ABS	ABS(integer-expression): Remove sign from signed value.
FLOOR	FLOOR(integer-expression): Rounds real number down to nearest integer.
CEIL	CEIL(real-expression): Rounds real number up to nearest integer.
MIN	MIN(integer-expression-list): Picks minimum integer or real value of comma separated list.
MAX	MAX(integer-expression-list): Picks maximum integer or real value of comma separated list.
COUNT	COUNT(integer-expression): Returns the number of binary 1's in the integer.

**Table 2: Functions**

Function	Definition
ROUND	ROUND(real-expression): Rounds to the nearest integer; halfway rounds away from zero.
UNIT	UNIT(fieldName UnitOfMeasure): Input operand is a register field name that defines all values with the same unit of measure. Returns the value expressed in the unit of measure for the current value of the register field.
POW	POW(base, exponent): POW(x,y) returns the value x to the power of y.

### 1.3.3 Operator Precedence and Associativity

This document follows C operator precedence and associativity. The following table lists operator precedence (highest to lowest). Their associativity indicates in what order operators of equal precedence in an expression are applied. Parentheses are also used to group sub-expressions to force a different precedence; such parenthetical expressions can be nested and are evaluated from inner to outer. E.g. “X = A | ~B & C” is the same as “X = A | ((~B) & C)”.

**Table 3: Operator Precedence and Associativity**

Operator	Description	Associativity
!, ~	Logical negation/bitwise complement	right-to-left
*, /, %	Multiplication/division/modulus	left-to-right
+, -	Addition/subtraction	left-to-right
<<, >>	Bitwise shift left, Bitwise shift right	left-to-right
<, <=, >, >=, ==, !=	Relational operators	left-to-right
&	Bitwise AND	left-to-right
^	Bitwise exclusive OR	left-to-right
	Bitwise inclusive OR	left-to-right
&&	Logical AND	left-to-right
	Logical OR	left-to-right
?:	Ternary conditional	right-to-left

## 1.4 Definitions

**Table 4: Definitions**

Term	Definition
AP	Application processor. See <a href="#">2.3 [Processor Initialization]</a> .
BAPM	Bidirectional Application Power Management. See <a href="#">2.5.9.3 [Bidirectional Application Power Management (BAPM)]</a> .
Battery-Power	The system is running from a battery power source or otherwise undocked from a continuous power supply. Setting using this definition may be required to change during runtime.
BCS	Base configuration space. See <a href="#">2.7 [Configuration Space]</a> .

Table 4: Definitions

Term	Definition
<b>BERT</b>	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.
<b>BIST</b>	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).
<b>Boot VID</b>	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. See <a href="#">2.5.1.2 [Internal VID Registers and Encodings]</a> .
<b>BCD</b>	Binary coded decimal number format.
<b>BSC</b>	Boot strap core. Core 0 of the <a href="#">BSP</a> . Specified by <a href="#">MSR0000_001B[BSC]</a> .
<b>BSP</b>	Boot strap processor. See <a href="#">2.3 [Processor Initialization]</a> .
<b>CAR</b>	Use of the L2 cache as RAM during boot. See <a href="#">2.3.3 [Using L2 Cache as General Storage During Boot]</a> .
<b>C-states</b>	These are ACPI-defined core power states. C0 is operational. All other C-states are low-power states in which the processor is not executing code. See <a href="#">2.5.3.2 [Core C-states]</a> .
<b>Canonical address</b>	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.
<b>Channel</b>	See DRAM channel.
<b>CMP</b>	Chip multi-processing. Refers to processors that include multiple cores. See <a href="#">2.1 [Processor Overview]</a> .
<b>COF</b>	Current operating frequency of a given clock domain. See <a href="#">2.5.3 [CPU Power Management]</a> .
<b>Cold reset</b>	PWROK is deasserted and RESET_L is asserted. See <a href="#">2.3 [Processor Initialization]</a> .
<b>Core Cluster</b>	Four <a href="#">Cores</a> that share L2 resources. See <a href="#">L2 complex</a> .
<b>L2 complex</b>	Four <a href="#">Cores</a> that share L2 resources. See <a href="#">2.1 [Processor Overview]</a> .
<b>Core</b>	The instruction execution unit of the processor. See <a href="#">2.1 [Processor Overview]</a> .
<b>CPB</b>	Core performance boost. See <a href="#">2.5.9.1 [Core Performance Boost (CPB)]</a> .
<b>CpuCore-Num</b>	Specifies the core number. See <a href="#">2.4.4 [Processor Cores and Downcoring]</a> .
<b>CPUID function X</b>	Refers to the CPUID instruction when EAX is preloaded with X. See <a href="#">3.17 [CPUID Instruction Registers]</a> .
<b>CS</b>	Chip select. See <a href="#">D18F2x[5C:40]_dct[0] [DRAM CS Base Address]</a> .
<b>DCT</b>	DRAM controller. See <a href="#">2.9 [DRAM Controllers (DCTs)]</a> .
<b>DCQ</b>	DRAM controller queue.
<b>DDR3</b>	DDR3 memory technology. See <a href="#">2.9 [DRAM Controllers (DCTs)]</a> .
<b>DID</b>	Divisor identifier. Specifies the post-PLL divisor used to reduce the COF. See <a href="#">2.5.3 [CPU Power Management]</a> .
<b>Doubleword</b>	A 32-bit value.
<b>Downcoring</b>	Removal of cores. See <a href="#">2.4.4 [Processor Cores and Downcoring]</a> .
<b>DRAM channel</b>	The part of the DRAM interface that connects to a DIMM. See <a href="#">2.9 [DRAM Controllers (DCTs)]</a> .

Table 4: Definitions

Term	Definition
<b>Dual-Plane</b>	Refers to a processor or systemboard where VDD and VDDNB are separate and may operate at independent voltage levels. Refer to 2.5.1 [Processor Power Planes And Voltage Control].
<b>DW</b>	Doubleword. A 32-bit value.
<b>ECS</b>	Extended configuration space. See 2.7 [Configuration Space].
<b>EDS</b>	Electrical data sheet. See 1.2 [Reference Documents].
<b>FCH</b>	Fusion Controller Hub. The integrated platform device that contains the IO subsystem and the bridge to system BIOS. See 2.15 [Fusion Controller Hub].
<b>FDS</b>	Functional data sheet; there is one FDS for each package type.
<b>FID</b>	Frequency identifier. Specifies the PLL frequency multiplier for a given clock domain. See 2.5.3 [CPU Power Management].
<b>FreeR-unSample-Timer</b>	An internal free running timer used by many power management features. The timer increments at the rate specified by D18F4x110[CSampleTimer].
<b>GB</b>	Gbyte or Gigabyte; 1,073,741,824 bytes.
<b>#GP</b>	A general-protection exception.
<b>#GP(0)</b>	Notation indicating a general-protection exception (#GP) with error code of 0.
<b>GpuEnabled</b>	GpuEnabled = (D1F0x00!=FFFF_FFFFh).
<b>GT/s</b>	Giga-transfers per second.
<b>HCD</b>	Host Controller Driver. A software component.
<b>HTC</b>	Hardware thermal control. See 2.10.3.1 [PROCHOT_L and Hardware Thermal Control (HTC)].
<b>HTC-active state</b>	Hardware-controlled lower-power, lower-performance state used to reduce temperature. See 2.10.3.1 [PROCHOT_L and Hardware Thermal Control (HTC)].
<b>IBS</b>	Instruction based sampling. See 2.6.2 [Instruction Based Sampling (IBS)].
<b>IFCM</b>	Isochronous flow-control mode, as defined in the link specification.
<b>ILM</b>	Internal loopback mode. Mode in which the link receive lanes are connected directly to the transmit lanes of the same link for testing and characterization. See D18F0x[18C:170] [Link Extended Control].
<b>IO configuration</b>	Access to configuration space through IO ports CF8h and CFCh. See 2.7 [Configuration Space].
<b>IORR</b>	IO range register. See MSRC001_00[18,16] [IO Range Base (IORR_BASE[1:0])].
<b>KB</b>	Kbyte or Kilobyte; 1024 bytes.
<b>L1 cache</b>	The level 1 caches (instruction cache and the data cache) and the level 2 caches. See 2.1 [Processor Overview].
<b>L2 cache</b>	
<b>L2I</b>	The L2 Interface complex common to a set of cores.
<b>L2BC</b>	L2I Base Core. The lowest numbered core for a L2I instance.
<b>Linear (virtual) address</b>	The address generated by a core after the segment is applied.
<b>Link</b>	Generic term that refers to a refer to PCIe® link.
<b>LINT</b>	Local interrupt.



Table 4: Definitions

Term	Definition
<b>Logical address</b>	The address generated by a core before the segment is applied.
<b>LVT</b>	Local vector table. A collection of APIC registers that define interrupts for local events. E.g., <a href="#">APIC[530:500] [Extended Interrupt [3:0] Local Vector Table]</a> .
<b>Master abort</b>	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.
<b>MB</b>	Megabyte; 1024 KB.
<b>MCT</b>	Memory controller. See <a href="#">2.8 [Northbridge (NB)]</a> .
<b>MCQ</b>	Memory controller queue. See <a href="#">2.8 [Northbridge (NB)]</a> .
<b>Micro-op</b>	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. Processor schedulers subsequently break down macro-ops into sequences of even simpler instructions called micro-ops, each of which specifies a single primitive operation. See <i>Software Optimization Guide for AMD Family 16h Processors</i> .
<b>MEMCLK</b>	Refers to the clock signals, M[B, A][3:0]_CLK, that are driven from the processor to DDR DIMMs.
<b>MMIO</b>	Memory-mapped input-output range. This is physical address space that is mapped to the IO functions such as the IO links or MMIO configuration. The IO link MMIO ranges are specified by <a href="#">D18F1x[2CC:2A0,1CC:180,BC:80] [MMIO Base/Limit]</a> .
<b>MMIO configuration</b>	Access to configuration space through memory space. See <a href="#">2.7 [Configuration Space]</a> .
<b>MSR</b>	Model-specific register. The core includes several MSRs for general configuration and control. See <a href="#">3.18 [MSRs - MSR0000_xxxx]</a> for the beginning of the MSR register definitions.
<b>MTRR</b>	Memory-type range register. The MTRRs specify the type of memory associated with various memory ranges. See <a href="#">MSR0000_00FE</a> , <a href="#">MSR0000_020[F:0]</a> , <a href="#">MSR0000_02[6F:68,59:58,50]</a> , and <a href="#">MSR0000_02FF</a> .
<b>NB</b>	Northbridge. The transaction routing block of the node. See <a href="#">2.1 [Processor Overview]</a> .
<b>NBC</b>	NBC = (CPUID Fn0000_0001_EBX[LocalApicId[3:0]]==0). Node Base Core. The lowest numbered core in the node.
<b>NBPMC</b>	Performance monitor counter. See <a href="#">2.6.1.3 [NB Performance Monitor Counters]</a> .
<b>NCLK</b>	The main northbridge clock. The NCLK frequency is the NB COF.
<b>Node</b>	See <a href="#">2.1 [Processor Overview]</a> .
<b>Normalized address</b>	Addresses used by DCTs. See <a href="#">2.8 [Northbridge (NB)]</a> .
<b>OW</b>	Octword. An 128-bit value.
<b>ODM</b>	On-DIMM mirroring. See <a href="#">D18F2x[5C:40]_dct[0][OnDimmMirror]</a> .
<b>ODT</b>	On-die termination, which is applied DRAM interface signals.
<b>ODTS</b>	DRAM On-die thermal sensor.
<b>Operational frequency</b>	The frequency at which the processor operates. See <a href="#">2.5 [Power Management]</a> .

Table 4: Definitions

Term	Definition
<b>PCIe®</b>	PCI Express®.
<b>PDS</b>	Product data sheet.
<b>Physical address</b>	Addresses used by cores in transactions sent to the NB.
<b>PMC</b>	Performance monitor counter. See 2.6.1.1 [Core Performance Monitor Counters].
<b>PRBS</b>	Pseudo-random bit sequence.
<b>Processor</b>	See 2.1 [Processor Overview].
<b>PSI</b>	Power Status Indicator. See 2.5.1.3.1 [PSIx_L Bit].
<b>P-state</b>	Performance state. See 2.5 [Power Management].
<b>PTE</b>	Page table entry.
<b>QW</b>	Quadword. A 64-bit value.
<b>RAS</b>	Reliability, availability and serviceability (industry term). See 2.14.1 [Machine Check Architecture].
<b>RDQ</b>	Read data queue.
<b>REFCLK</b>	Reference Clock, refers to the clock frequency (100 MHz) or the clock period (10 ns) depending on the context used.
<b>RX</b>	Receiver.
<b>RevA0</b>	RevA0 = ((D18F3xFC[BaseModel] == 0h) && (D18F3xFC[Stepping] == 0h))
<b>RevA1</b>	RevA1 = ((D18F3xFC[BaseModel] == 0h) && (D18F3xFC[Stepping] == 1h)).
<b>Scrubber</b>	Background memory checking logic. See 2.8.3 [Memory Scrubbers].
<b>Shutdown</b>	A state in which the affected core waits for either INIT, RESET, or NMI. When shutdown state is entered, a shutdown special cycle is sent on the IO links.
<b>Single-Plane</b>	Refers to a processor or systemboard where VDD and VDDNB are tied together and operate at the same voltage level. Refer to 2.5.1 [Processor Power Planes And Voltage Control].
<b>Slam</b>	Refers to changing the voltage to a new value in one step (as opposed to stepping). See 2.5.1.4.1 [Hardware-Initiated Voltage Transitions].
<b>SMAF</b>	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 D18F3x[84:80] [ACPI Power State Control].
<b>SMBus</b>	System management bus. Refers to the protocol on which the serial VID interface (SVI) commands are based. See 2.5.1 [Processor Power Planes And Voltage Control], and 1.2 [Reference Documents].
<b>SMC</b>	System management controller. This is the platform device that communicates system management state information to the processor through an IO link, typically the system IO hub.
<b>SMI</b>	System management interrupt. See 2.4.8.2.1 [SMM Overview].
<b>SMM</b>	System management mode. See 2.4.8.2 [System Management Mode (SMM)].
<b>Speculative event</b>	A performance monitor event counter that counts all occurrences of the event even if the event occurs during speculative code execution.
<b>SVI2</b>	Serial VID 2.0 interface. See 2.5.1.1 [Serial VID Interface]
<b>SVM</b>	Secure virtual machine. See 2.4.9 [Secure Virtual Machine Mode (SVM)].

**Table 4: Definitions**

Term	Definition
<b>Sync flood</b>	The propagation of continuous sync packets to all links. This is used to quickly stop the transmission of potentially bad data when there are no other means to do so. See the link specification for additional information.
<b>TCC</b>	Temperature calculation circuit. See <a href="#">2.10 [Thermal Functions]</a> .
<b>Tctl</b>	Processor temperature control value. See <a href="#">2.10.3 [Temperature-Driven Logic]</a> .
<b>TDC</b>	Thermal design current. See the AMD Infrastructure Roadmap, #41482.
<b>TDP</b>	Thermal design power. A power consumption parameter that is used in conjunction with thermal specifications to design appropriate cooling solutions for the processor. See <a href="#">2.5.9.2 [TDP Limiting]</a> .
<b>Token</b>	A scheduler entry used in various northbridge queues to track outstanding requests. See <a href="#">D18F3x140 [SRI to XCS Token Count]</a> on Page 461.
<b>TX</b>	Transmitter.
<b>UI</b>	Unit interval. This is the amount of time equal to one half of a clock cycle.
<b>UMI</b>	Unified Media Interface. The link between the processor and the FCH.
<b>VDD</b>	Main power supply to the processor core logic.
<b>VDDNB</b>	Main power supply to the processor NB logic.
<b>VID</b>	Voltage level identifier. See <a href="#">2.5.1 [Processor Power Planes And Voltage Control]</a> .
<b>Virtual CAS</b>	The clock in which CAS is asserted for the burst, N, plus the burst length (in MEMCLKs), minus 1; so the last clock of virtual CAS = N + (BL/2) - 1.
<b>VRM</b>	Voltage regulator module.
<b>W</b>	Word. A 16-bit value.
<b>Warm reset</b>	RESET_L is asserted only (while PWROK stays high). See <a href="#">2.3 [Processor Initialization]</a> .
<b>WDT</b>	Watchdog timer. A timer that detects activity and triggers an error if a specified period of time expires without the activity. For example, see <a href="#">MSRC001_0074 [CPU Watchdog Timer (Cpu-WdtCfg)]</a> or the NB watchdog timer in <a href="#">D18F3x40 [MCA NB Control]</a> .
<b>WDQ</b>	Write data queue.
<b>XBAR</b>	Cross bar; command packet switch. See <a href="#">2.8 [Northbridge (NB)]</a> .

## 1.5 Changes Between Revisions and Product Variations

### 1.5.1 Revision Conventions

The processor revision is specified by [CPUID Fn0000\\_0001\\_EAX \[Family, Model, Stepping Identifiers\]](#) or [CPUID Fn8000\\_0001\\_EAX \[Family, Model, Stepping Identifiers\]](#). This document uses a revision letter instead of specific model numbers. The following table contains the definitions based on model and stepping used in this document. Where applicable, the processor stepping is indicated after the revision letter. All behavior marked with a revision letter apply to future revisions unless they are superseded by a change in a later revision. See the revision guide for additional information about revision determination. See [1.2 \[Reference Documents\]](#).

**Table 5: Processor revision conventions**

Term	Definition
<b>PROC</b>	Processor. PROC = {CPUID Fn0000_0001_EAX[ExtFamily], {CPUID Fn0000_0001_EAX[ExtModel], CPUID Fn0000_0001_EAX[BaseModel]}, CPUID Fn0000_0001_EAX[Stepping]}.
<b>KB_A1</b>	KB_A1 = {07h,00h,1h}.

### 1.5.2 Major Changes

This section describes the major changes relative to Family 14h Models 00h-0Fh (ON) Processors.

- CPU core changes:
  - Architectural changes:
    - Shared L2 Cache.
    - MSRC001\_023[6,4,2,0], MSRC001\_023[7,5,3,1] L2I Performance counters indicated by CPUID Fn8000\_0001\_ECX[PerfCtrExtL2I].
    - MSRC001\_024[6,4,2,0], MSRC001\_024[7,5,3,1]: NB Performance counters indicated by CPUID Fn8000\_0001\_ECX[PerfCtrExtNB].
    - CPUID Fn8000\_0008\_EAX[PhysAddrSize]: Physical Address Extended to 40 bits.
    - CPUID Fn8000\_0001\_ECX[DataBreakpointExtension]: Debug Breakpoint Extension.
  - Instruction set changes:
    - CPUID Fn0000\_0001\_ECX[AVX]: Added AVX instruction support.
    - CPUID Fn0000\_0001\_ECX[XSAVE,OSXSAVE]: CPUID Fn0000\_000D: Added XSAVE support.
    - CPUID Fn0000\_0001\_ECX[AES]: Added AES instruction support.
    - CPUID Fn0000\_0001\_ECX[SSE41, SSE42]: Added SSE4.1 and SSE4.2 instruction support.
    - CPUID Fn0000\_0001\_ECX[PCLMULQDQ]: Added PCLMULQDQ instruction support.
    - CPUID Fn0000\_000D\_EAX\_x0-CPUID Fn0000\_000D\_EDX\_x2: Added XSAVE/XRSTOR.
    - CPUID Fn0000\_0001\_ECX[MOVBE]: Added MOVBE instruction support.
    - CPUID Fn0000\_0001\_ECX[F16C]: Added F16C instruction support.
    - CPUID Fn0000\_0007\_EBX\_x0[BMI1]: Added BMI1 instruction support.
  - Virtualization Support
    - CPUID Fn8000\_000A\_EDX[TscRateMsr]: Added TSC Scaling.
    - CPUID Fn8000\_000A\_EDX[FlushByAsid]: Added Flush by ASID.
    - CPUID Fn8000\_000A\_EDX[DecodeAssists]: Added Decode Assist.
    - CPUID Fn8000\_000A\_EDX[PauseFilterThreshold]: Added Enhanced Pause Filter.

#### 1.5.2.1 Major Changes to Core/NB Performance Counters

Major Changes to Core/NB Performance Counters:

PerfMon Changes: Added L2I PerfMon group. See MSRC001\_023[6,4,2,0] and MSRC001\_023[7,5,3,1].

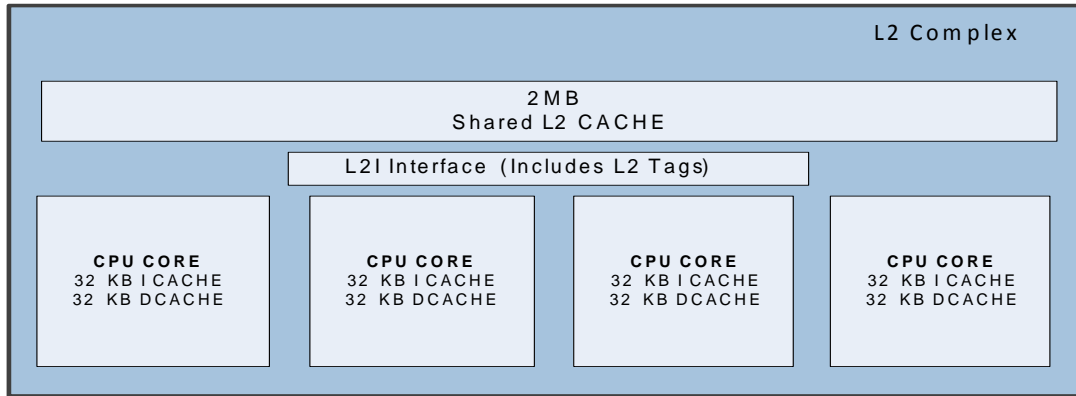
## 2 Functional Description

### 2.1 Processor Overview

The *processor* is defined as follows:

- The processor is a package that contains one node.
- Supports x86-based instruction sets.
- Packages:

- FT3: Notebook Package.
- FS1b: Desktop Package.
- See [CUID Fn8000\\_0001\\_EBX\[PkgType\]](#).
- L2 complex



**Figure 1: A L2 Complex**

- 1 L2 complex (4 cores)
- 2 MB L2
- See [2.4.1 \[L2 complex\]](#).
- DRAM:
  - One 64-bit DDR3 memory channel (A).
- Northbridge:
  - One communication packet routing block referred to as the northbridge (NB). The NB routes transactions between the cores, the link, and the DRAM interfaces. It includes the configuration register space for the device.
- Graphics northbridge:
  - Link:
    - PCIe® Gen2
    - One x4 Gfx link, four x1 GPP links.
    - See [2.11.3 \[Links\]](#).
- Power Management:
  - See [2.5 \[Power Management\]](#).
- RAS:
  - See [2.14 \[RAS Features\]](#).

## 2.2 System Overview

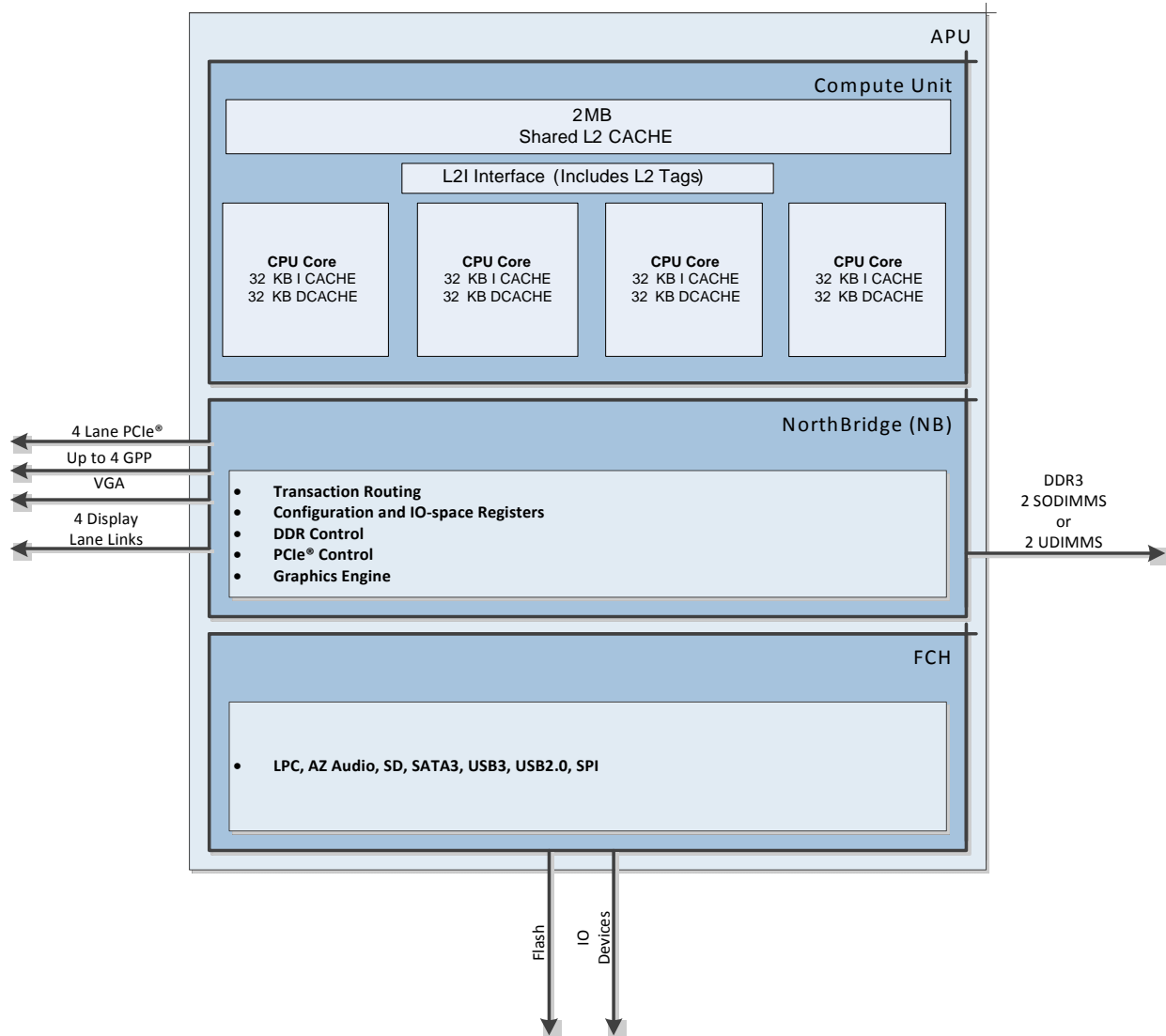


Figure 2: System Diagram

## 2.3 Processor Initialization

This section describes the initialization sequence after a cold reset.

Core 0 of the processor, the bootstrap core (BSC), begins executing code from the reset vector. The remaining cores do not fetch code until their enable bits are set (`D18F0x1DC[CpuEn]`).

### 2.3.1 BSC Initialization

The BSC must perform the following tasks as part of boot.

- Store BIST information from the EAX register into an unused processor register.

- **D18F0x6C**[InitDet] may be used by BIOS to differentiate between INIT and cold/warm reset.
- Determine type of startup using **D18F0x6C**[ColdRstDet].
  - If this is a warm reset then BIOS may check for valid MCA errors and if present save the status for later use. See 2.14.1.6 [Handling Machine Check Exceptions].
- Enable the cache, program the MTRRs for **CAR** and initialize **CAR**. See 2.3.3 [Using L2 Cache as General Storage During Boot].
- Setup the SMU.
- Setup of APIC (2.4.8.1.3 [ApicId Enumeration Requirements]).
- Setup the link configuration 2.11.3.2 [Link Configurations].
- Setup the root complex and initialize the I/O links 2.11.4.3 [Link Configuration and Initialization].
- If required, reallocate data and flow control buffers of the links (see **D18F0x[F0,D0,B0,90]** [Link Base Channel Buffer Count] and **D18F0x[F4,D4,B4,94]** [Link Isochronous Channel Buffer Count]).
- Issue system warm reset.
- Configure the DRAM controllers.
- Configure processor power management. See 2.5 [Power Management].
- Allow other cores to begin fetching instructions by setting **D18F0x1DC**[CpuEn] in the PCI configuration space of all nodes. See 2.4.4 [Processor Cores and Downcoring].

### 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 boot.

- Store BIST information from the **eax** register into an unused processor register.
- **D18F0x6C**[InitDet] may be used by BIOS to differentiate between INIT and cold/warm reset.
- Determine the history of this reset using the **D18F0x6C** [Link Initialization Control] [ColdRstDet] bit:
  - If this is a warm reset then BIOS may check for valid MCA errors and if present save the status for use later. See 2.14.1.6 [Handling Machine Check Exceptions].
- Set up the local APIC. See 2.4.8.1.3 [ApicId Enumeration Requirements].
- Configure processor power management. See 2.4 [Core].

### 2.3.3 Using L2 Cache as General Storage During Boot

Prior to initializing the DRAM controller for system memory, BIOS may use the L2 cache as general storage. If BIOS is not using L2 cache as general storage during boot, program **MSRC001\_102A**[CILinesToL2Dis]=0.

The L2 cache as general storage is described as follows:

- Each L2 complex 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=2MB) then there are 16 blocks of cache, each 2MB/16 in size, or 128KB each.
  - For each of the following values of L2Size, the following values are defined:
    - L2Size=1 MB: L2Tag=PhysAddr[39:16], L2WayIndex=PhysAddr[15:6].
    - L2Size=2 MB: 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:

- Picks an invalid way before picking a valid way.
- Prioritizes 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 1 MB and can rely on being able to use all 16 ways for general storage. See [CUID Fn8000\\_0006\\_ECX\[L2Size\]](#).

The following memory types are supported:

- **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.
  - WP-IO accesses, both read and write, do not get allocated into the L2 and therefore do not need to be considered for allocation into the L2.
- **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 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 is not allowed to deallocate a line in the L2 by using CLFLUSH.

Performance monitor event [L2IPMCx060\[3:2\]](#), titled “write victim block“, can be used to indicate whether L2 dirty data was lost by being 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\\_1021\[DisSpecTlbRld\]](#)=1. Disable speculative ITLB reloads.
- [MSRC001\\_1022\[DisSpecTlbWalk\]](#)=1. Disable speculative DTLB reloads.
- [MSRC001\\_1022\[DisHwPf\]](#)=1.
- [MSRC001\\_10A0\[L2RinserDis\]](#)=1.
- [MSRC001\\_10A0\[PrefetcherDis\]](#)=1. Disable L2 hardware prefetcher.
- [MSRC001\\_10A0\[CacheIcAttrDis\]](#)=1.
- CLFLUSH, INVD, and WBINVD must not be used during [CAR](#) but may be used when tearing down [CAR](#) for all compute units on a node.
- The BIOS must not use SSE, or MMX™ instructions, with the exception of the following list: MOVD, MOVQ, MOVQ2DQ, MOVQ2Q.
- The BIOS must not enable exceptions, page-faults, and other interrupts.
- BIOS must not use software prefetches.
- UC-DRAM: All of DRAM that is not accessed as WB-DRAM space must be marked as UC memory type.

When BIOS has completed using the cache for general storage the following steps are followed:



1. Prior to issuing an INVD instruction, BIOS must ensure that no other source of coherent traffic, such as a DMA engine, is operating in the system. This condition must hold true until the INVD instruction for tear-down has been executed on all CPUs.
2. An INVD instruction is executed on each core that used cache as general storage; an INVD is issued when all cores on all nodes have completed using the cache for general storage.
3. 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.
4. Program the following configuration state (Order is unimportant):
  - `MSRC001_0015[INVDWBINVD]=1.`
  - `MSRC001_1021[DisSpecTlbRld]=0.`
  - `MSRC001_1022[DisSpecTlbWalk]=0.`
  - `MSRC001_1022[DisHwPf]=0.`
  - `MSRC001_10A0[PrefetcherDis]=0.`
  - `MSRC001_10A0[CacheIcAttrDis]=0.`
  - `MSRC001_102A[CILinesToL2Dis]=0.`

### 2.3.4 Instruction Cache Configuration Register Usage Requirements

Modification of `MSRC001_1021 [Instruction Cache Configuration (IC_CFG)]` requires the following steps for all cores within the L2 complex:

- Synchronize all cores.
- Modify `MSRC001_1021` observing the **Same-for-all** requirement.
- Flush all caches using WBINVD for all cores within the L2 complex.
- Resume all cores.

## 2.4 Core

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

### 2.4.1 L2 complex

Each *L2 complex* includes 4 cores each having an x86 instruction execution logic and first-level (L1) data cache and first-level (L1) instruction cache. The second level (L2) general-purpose cache is shared between all cores of the L2 complex.

There is a set of MSRs and APIC registers associated with each core. Processors that include multiple cores are said to incorporate *chip multi-processing* or CMP. Unless otherwise specified the processor configuration interface hides the L2 complex implementation and presents software with homogenous cores, each independent of the other.

Software may use `D18F5x80[Enabled, DualCore, TripleCore, QuadCore]` in order to associate a core with a L2 complex. This information can be useful because some configuration settings are determined based on active L2 complexes and core performance may vary based on resource sharing within a L2 complex.

### 2.4.2 Caches and TLBs

Cache and TLB storage available to a core is reported by:

- `CPUID Fn8000_0005_EAX-CPUID Fn8000_0006_EDX.`
- `CPUID Fn8000_0019_EAX-CPUID Fn8000_0019_E[D,C]X.`
- `CPUID Fn8000_001D_EAX_x0-CPUID Fn8000_001E_EDX.`

Cache and TLB storage available to a core is summarized as follows:

- L1 and L2 Caches:
  - DC: 32 KB, 8-way, write-back, per-core.
  - IC: 32 KB, 2-way, per-core.
  - L2: 1 MB or 2 MB (Product-specific), 16-way associative, shared between all cores of a L2 complex.
- TLBs:
  - D, L1TLB:
    - 4 KB: 40 entries, fully associative.
    - 2 MB: 8 entries, fully associative.
  - D, L2TLB:
    - 4 KB: 512 entries, 4 way associative
    - 2 MB: 256 entries, hashed 2 way associative.
  - I, L1TLB:
    - 4 KB: 32 entries, fully associative.
    - 2 MB, 8 entries, fully associative.
  - I, L2TLB:
    - 4 KB: 512 entries, 4-way associative.

#### 2.4.2.1 Registers Shared by Cores in a L2 complex

Some registers are implemented one instance per L2 complex instead of per core; these registers are designated as [Per-L2](#). The absence of [Per-L2](#) implies the normal per-core instance programming model.

Programming rules for [Per-L2](#) registers:

- Software must ensure that a shared MSR written by one core on a L2 complex will not cause a problem for software that is running on the other core of the L2 complex.
- [Per-L2](#): A write to a [Per-L2](#) MSR does not have to be written to the other cores of the L2 complex in order for the other cores to see the updated value.
- A read-modify-write of a shared MSR register is not atomic. Software must ensure atomicity between the cores that could simultaneously read-modify-write the shared register.

#### 2.4.3 Virtual Address Space

The processor supports 48 address bits of virtual memory space (256 TB) as indicated by [CPUID Fn8000\\_0008\\_EAX](#).

#### 2.4.4 Processor Cores and Downcoring

Each L2 Complex supports downcoring as follows:

- The number of cores supported is specified by [D18F5x84\[CmpCap\]](#).
- The cores of a L2 complex may be software downcored by [D18F3x190\[DisCore\]](#). See 2.4.4.1 [[Software Downcoring using D18F3x190\[DisCore\]](#)].
  - Clocks are turned off and power is gated to downcored L2 complexes. The power savings is the same as CC6.
  - There must be at least 1 L2 complex enabled.
  - [D18F3x190\[DisCore\]](#) affects the value of [CPUID Fn0000\\_0001\\_EBX\[LogicalProcessorCount\]](#), [CPUID Fn0000\\_0001\\_EDX\[HTT\]](#), [CPUID Fn8000\\_0001\\_ECX\[CmpLegacy\]](#), [CPUID Fn8000\\_0008\\_ECX\[NC\]](#), [D18F5x80\[Enabled, DualCore, TripleCore, QuadCore\]](#). [D18F3x190\[DisCore\]](#) does not affect the value of [D18F5x84\[CmpCap\]](#).
- An implemented (physical) core that is downcored is not visible to software. Cores that are not downcored are numbered logically in a contiguous manner.

- **D18F5x80 [Compute Unit Status 1]** reports core topology information to software.
- The number of cores specified in **CPUID Fn8000\_0008\_ECX[NC]** must be the same as the number of cores enabled in **D18F0x1DC[CpuEn]**.
- The core number, *CpuCoreNum*, is provided to SW running on each core through **CPUID Fn0000\_0001\_EBX[LocalApicId]** and **APIC20[ApicId]**; *CpuCoreNum* also affects **D18F0x1DC[CpuEn]**. *CpuCoreNum*, varies as the lowest integers from 0 to **D18F5x84[CmpCap]**, based on the number of enabled cores; e.g., a 4-core node with 1 core disabled results in cores reporting *CpuCoreNum* values of 0, 1, and 2 regardless of which core is disabled. The boot core is always the core reporting *CpuCoreNum*=0.

Some legacy operating systems do not support processors with a non-power-of-2 number of cores. The BIOS is recommended to support a user configurable option to disable cores down to a power-of-2 number of cores for legacy operating system support.

#### 2.4.4.1 Software Downcoring using **D18F3x190[DisCore]**

Cores may be downcored by **D18F3x190[DisCore]**.

Software is required to use **D18F3x190[DisCore]** as follows:

- Setting bits corresponding to cores that are not present results in undefined behavior.
- Once a core has been removed by **D18F3x190[DisCore]=1**, it cannot be added back without a cold reset. E.g. Software may only set *DisCore* bits, never clear them.
- For enabled **L2 complexes** where (**L2 complex**>=1) software may not disable **Core 0** of the **L2 complex**.
- The most significant bit *N* is (the number of cores)-1 at cold reset; the number of cores at cold reset is (**CPUID Fn8000\_0008\_ECX[NC]**+1).
- The most significant bit *N* and the core ID significance of *DisCore* is not affected by the value of *DisCore* followed by a warm-reset.
  - E.g. If core 2 is disabled by **DisCore[3:0]=0100b** followed by a warm reset, then the new core 2 is the old core 3. If the new core 2 needs to then be disabled then **DisCore[3:0]=1100b** followed by a warm reset.
- All bits greater than bit *N* are reserved.
- If **D18F3x190[DisCore]** is changed, then the following need to be updated:
  - **D18F0x60[CpuCnt[4:0]]**.
  - **D18F5x170[NbPstateThreshold]**

#### 2.4.5 Physical Address Space

The processor supports a 40 bit physical address space, as indicated by **CPUID Fn8000\_0008\_EAX [Long Mode Address Size Identifiers]**.

The processor master aborts the following upper-address transactions (to address *PhysAddr*):

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

#### 2.4.6 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.

##### 2.4.6.1 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. An IO link access to physical address space indicates to the NB the cache attribute (Coherent or Non-coherent, based on bit[0] of the Sized Read and Write commands).

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

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

#### 2.4.6.1.1 Determining Memory Type

The memory type for a core access is determined by the highest priority of the following ranges that the access falls in: 1==Lowest priority.

1. The memory type as determined by architectural mechanisms.
  - See the APM2 chapter titled “Memory System”, sections “Memory-Type Range Registers” and “Page-Attribute Table Mechanism”.
  - See the APM2 chapter titled “Nested Paging”, section “Combining Memory Types, MTRRs”.
  - See [MSR0000\\_02FF \[MTRR Default Memory Type \(MTRRdefType\)\]](#), [MSR0000\\_020\[F:0\] \[Variable-Size MTRRs Base/Mask\]](#), [MSR0000\\_02\[6F:68,59:58,50\] \[Fixed-Size MTRRs\]](#).
2. TSeg & ASeg SMM mechanism. (see [MSRC001\\_0112](#) and [MSRC001\\_0113](#))
3. CR0[CD]: If (CR0[CD]==1) then MemType=CD.
4. MMIO config space, APIC space.
  - MMIO APIC space and MMIO config space must not overlap.
  - MemType=UC.
  - See [2.4.8.1.2 \[APIC Register Space\]](#) and [2.7 \[Configuration Space\]](#).
5. If (“In SMM Mode” && ~(([MSRC001\\_0113\[AValid\]](#) && “The address falls within the ASeg region”) || ([MSRC001\\_0113\[TValid\]](#) && “The address falls within the TSeg region”))) then MemType=CD.

#### 2.4.6.1.2 Determining The Access Destination for Core 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.

1. RdDram/WrDram as determined by [MSRC001\\_001A \[Top Of Memory \(TOP\\_MEM\)\]](#) and [MSRC001\\_001D \[Top Of Memory 2 \(TOM2\)\]](#).
2. The IORRs. (see [MSRC001\\_00\[18,16\]](#) and [MSRC001\\_00\[19,17\]](#)).
3. The fixed MTRRs. (see [MSR0000\\_02\[6F:68,59:58,50\] \[Fixed-Size MTRRs\]](#))
4. TSeg & ASeg SMM mechanism. (see [MSRC001\\_0112](#) and [MSRC001\\_0113](#))
5. MMIO config space, APIC space.
  - MMIO APIC space and MMIO config space must not overlap.
  - RdDram=IO, WrDram=IO.
  - See [2.4.8.1.2 \[APIC Register Space\]](#) and [2.7 \[Configuration Space\]](#).
6. NB address space routing. See [2.8.2.1.1 \[DRAM and MMIO Memory Space\]](#).

#### 2.4.7 Timers

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

- [MSR0000\\_0010 \[Time Stamp Counter \(TSC\)\]](#); the TSC increments at the rate specified by the P0 P-state.

- See 2.5.3.1.1.1 [Software P-state Numbering].
- See MSRC001\_00[6B:64] [P-state [7:0]].
- The APIC timer (APIC380 and APIC390), which increments at the rate of 2xCLKIN; the APIC timer may increment in units of between 1 and 8.

## 2.4.8 Interrupts

### 2.4.8.1 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 IO hub (IO APICs)
- Other local APICs (inter-processor interrupts)
- APIC timer
- Thermal events
- Performance counters
- Legacy local interrupts from the IO hub (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.

#### 2.4.8.1.1 Detecting and Enabling

APIC is detected and enabled via CPUID Fn0000\_0001\_EDX[APIC].

The local APIC is enabled via MSR0000\_001B[ApicEn]. Reset forces APIC disabled.

#### 2.4.8.1.2 APIC Register Space

MMIO APIC space:

- Memory mapped to a 4 KB range. The memory type of this space is the UC memory type. The base address of this range is specified by {MSR0000\_001B[ApicBar[47:12]], 000h}.
- The mnemonic is defined to be APICXX; XX is the byte address offset from the base address.
- MMIO APIC registers in xAPIC mode is defined by the register from APIC20 to APIC[530:500].
- Treated as normal memory space when APIC is disabled, as specified by MSR0000\_001B[ApicEn].

#### 2.4.8.1.3 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 in the processor is identical. See 2.4.10.1 [Multi-Core Support] to derive NC.

Operating systems are expected to use CPUID Fn8000\_0008\_ECX[ApicIdCoreIdSize], 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].)  $MNC = (2 \wedge CPUID Fn8000_0008_ECX[ApicIdCoreIdSize])$ . BIOS must use the `ApicId MNC` rule when assigning `APIC20[ApicId]` values as described below.

`ApicId MNC` rule: The `ApicId` of core `j` on processor `i` must be enumerated/assigned as:

- `ApicId[proc=i, core=j] = (OFFSET_IDX + i) * MNC + j`
- Where `OFFSET_IDX` is an integer offset (0 to N) used to shift up the core `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  $N = (\text{Number\_Of\_Processors} * MNC)$  and  $M = \text{Number\_Of\_IOAPICs}$ :

- If  $(N+M) < 16$ , then assign the local (core) `ApicIds` first from 0 to  $N-1$ , and the IOAPIC IDs from  $N$  to  $N+(M-1)$ . APIC ID 15 is reserved for broadcast when `APIC410[ExtApicIdEn]==0`.
- If  $(N+M) \geq 16$ , then assign the IOAPIC IDs first from 0 to  $M-1$ , and the local (core) `ApicIds` from  $K$  to  $K+(N-1)$ , where  $K$  is an integer multiple of  $MNC$  greater than  $M-1$ .

#### 2.4.8.1.4 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.

#### 2.4.8.1.5 Logical Destination Mode

A local APIC accepts interrupts selected by `APICD0 [Logical Destination (LDR)]` 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.

#### 2.4.8.1.6 Interrupt Delivery

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

When an APIC accepts a non-vectorized 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 `APIC[270:200] [Interrupt Request (IRR)]` 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. The corresponding bit in `APIC[1F0:180] [Trigger Mode (TMR)]` is set if the interrupt is level-triggered and cleared if edge-triggered. 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.

#### 2.4.8.1.7 Vectored Interrupt Handling

APIC80 [Task Priority (TPR)] and APICA0 [Processor Priority (PPR)] 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 APIC[170:100] [In-Service (ISR)]. 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][Request-Bits]) 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].

When the processor has completed service for an interrupt, it performs a write to APICB0 [End of Interrupt], 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.

#### 2.4.8.1.8 Interrupt Masking

Interrupt masking is controlled by the APIC410 [Extended APIC Control]. If APIC410[IerEn] is set, APIC[4F0:480] [Interrupt Enable] 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.

#### 2.4.8.1.9 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 APICF0 [Spurious-Interrupt Vector (SVR)]. APIC[170:100] is not changed and no write to APICB0 occurs.

#### 2.4.8.1.10 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.

#### 2.4.8.1.11 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[IerEn]` 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 `APIC90 [Arbitration Priority (APR)]`, 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 in `APIC[170:100][InServiceBits]` (ISRVec). If `APIC410[IerEn]` 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]) && (APIC80[Priority[7:4]] > ISRVec[7:4])) THEN
    APIC90[Priority] = APIC80[Priority]
ELSEIF (IRRVec[7:4] > ISRVec[7:4]) THEN
    APIC90[Priority] = {IRRVec[7:4],0h}
ELSE
    APIC90[Priority] = {ISRVec[7:4],0h}
ENDIF.
```

#### 2.4.8.1.12 Inter-Processor Interrupts

`APIC300 [Interrupt Command Low (ICR Low)]` and `APIC310 [Interrupt Command High (ICR High)]` 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.

#### 2.4.8.1.13 APIC Timer Operation

The local APIC contains a 32-bit timer, controlled by `APIC320 [LVT Timer]`, `APIC380 [Timer Initial Count]`, and `APIC3E0 [Timer Divide Configuration]`. 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 `APIC390 [Timer Current Count]`. `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.

#### 2.4.8.1.14 Generalized Local Vector Table

All LVTs (`APIC330` to `APIC3[60:50]`, and `APIC[530:500]`) support a generalized message type as follows:

- 000b=Fixed
- 010b=SMI
- 100b=NMI
- 111b=ExtINT



- All other messages types are reserved.

#### 2.4.8.1.15 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]` is unaffected.
- Pending APIC register writes complete.

#### 2.4.8.2 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.

##### 2.4.8.2.1 SMM Overview

SMM is entered by a core on the next instruction boundary after a system management interrupt (SMI) is received and recognized. A core 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.

##### 2.4.8.2.2 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 jump, call or return in the SMI handler can only address the lower 1M of memory, unless the SMI handler first switches to protected mode.
  - If (`MSRC001_0111[SmmBase]>=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.
- 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 `MSRC001_0111[SmmBase]`. Important offsets to the base address pointer are:

- `MSRC001_0111[SmmBase] + 8000h`: SMI handler entry point.
- `MSRC001_0111[SmmBase] + FE00h - FFFFh`: SMM state save area.

#### 2.4.8.2.3 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 IO hub and trigger a global SMI interrupt message back to the coherent fabric.

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

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

The status for these is stored in `SMMFEC4`.

#### 2.4.8.2.4 SMM Initial State

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

**Table 6: SMM Initial State**

Register	SMM Initial State
CS	SmmBase[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

**Table 6: SMM Initial State**

Register	SMM Initial State
TR	Unmodified
DR6	Unmodified
DR7	0000_0000_0000_0400h
EFER	All bits are cleared except bit 12 (SVME) which is unmodified.

**2.4.8.2.5 SMM Save State**

In the following table, the offset field provides the offset from the SMM base address specified by [MSRC001\\_0111 \[SMM Base Address \(SMM\\_BASE\)\]](#).

**Table 7: 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	
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} <sup>1</sup>	
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} <sup>1</sup>	
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	

Table 7: SMM Save State

Offset	Size	Contents	Access
FE80h	4 Bytes	IDTR	Read-only
FE84h	Word	Reserved	
FE88h	2 Bytes	Limit	
FE90h	Quadword	Reserved	
FE90h	Word	TR	Read-only
FE92h	Word	Selector	
FE94h	Doubleword	Attributes	
FE98h	Quadword	Limit	
FEA0h	Quadword	Base	Read-only
FEA8h	Quadword	IO_RESTART_RIP	
FEB0h	Quadword	IO_RESTART_RCX	
FEB8h	Quadword	IO_RESTART_RSI	
FEC0h	Doubleword	IO_RESTART_RDI	Read-only
FEC4	Doubleword	SMMFEC0 [SMM IO Trap Offset]	Read-only
FEC8h	Byte	SMMFEC4 [Local SMI Status]	Read-write
FEC9h	Byte	SMMFEC8 [SMM IO Restart Byte]	Read-write
FECAh	Byte	SMMFEC9 [Auto Halt Restart Offset]	Read-write
FECBh	5 Bytes	SMMFECA [NMI Mask]	
FED0h	Quadword	Reserved	Read-only
FED8h	Quadword	EFER	Read-only
FEE0h	Quadword	SMMFED8 [SMM SVM State]	Read-only
FEE8h	Quadword	Guest VMCB physical address	Read-only
FEF0h	16 Bytes	SVM Virtual Interrupt Control	Read-only
FEFCh	Doubleword	Reserved	Read-only
FF00h	Doubleword	SMMFEFC [SMM-Revision Identifier]	Read-write
FF04h	28 Bytes	SMMFF00 [SMM Base Address (SMM_BASE)]	
FF20h	Quadword	Reserved	Read-only
FF28h	Quadword	Guest PAT	
FF30h	Quadword	Host EFER <sup>2</sup>	
FF38h	Quadword	Host CR4 <sup>2</sup>	
FF40h	Quadword	Nested CR3 <sup>2</sup>	
FF48h	Quadword	Host Cr0 <sup>2</sup>	
FF50h	Quadword	CR4	
FF58h	Quadword	CR3	
FF60h	Quadword	CR0	
FF68h	Quadword	DR7	Read-only
FF70h	Quadword	DR6	
		RFLAGS	Read-write

**Table 7: SMM Save State**

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

Notes:

1. This notation specifies that bit[47] is replicated in each of the 16 MSBs of the DW (sometimes called *sign extended*). The 16 LSB's contain bits[47:32].
2. Only used 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, [SMMFEC0 \[SMM IO Trap Offset\]](#) 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 [SMMFEC8 \[SMM IO Restart Byte\]](#), to cause the core to re-execute the IO instruction immediately after resuming from SMM.

Bits	Description
31:16	<b>Port: trapped IO port address.</b> Read-only. This provides the address of the IO instruction.
15:12	<b>BPR: IO breakpoint match.</b> Read-only.
11	<b>TF: EFLAGS TF value.</b> Read-only.

10:7	Reserved.
6	<b>SZ32: size 32 bits.</b> Read-only. 1=Port access was 32 bits.
5	<b>SZ16: size 16 bits.</b> Read-only. 1= Port access was 16 bits.
4	<b>SZ8: size 8 bits.</b> Read-only. 1=Port access was 8 bits.
3	<b>REP: repeated port access.</b> Read-only.
2	<b>STR: string-based port access.</b> Read-only.
1	<b>V: IO trap word valid.</b> 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	<b>RW: port access type.</b> 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	<b>SmiSrcLvtExt: SMI source LVT extended entry.</b> Read-only. This bit is associated with the SMI sources specified in <a href="#">APIC[530:500] [Extended Interrupt [3:0] Local Vector Table]</a> .
16	<b>SmiSrcLvtLcy: SMI source LVT legacy entry.</b> Read-only. This bit is associated with the SMI sources specified by the non-extended LVT entries of the APIC.
15:9	Reserved.
8	<b>MceRedirSts: machine check exception redirection status.</b> Read-only. This bit is associated with the SMI source specified in <a href="#">MSRC001_0022[RedirSmiEn]</a> .
7:4	Reserved.
3:0	<b>IoTrapSts: IO trap status.</b> Read-only. Each of these bits is associated with each of the respective SMI sources specified in <a href="#">MSRC001_00[53:50] [IO Trap (SMI_ON_IO_TRAP_[3:0])]</a> .

### 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](#) field 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 core services the second SMI prior to re-executing the trapped IO instruction. [SMMFEC0](#) field 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 [SMMFEC8 \[SMM IO Restart Byte\]](#), 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	<b>RST: SMM IO Restart Byte.</b> Read-write.

### SMMFEC9 Auto Halt Restart Offset

Bits	Description
7:1	Reserved.
0	<b>HLT: halt restart.</b> Read-write. 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. 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.

### SMMFECA NMI Mask

Bits	Description
7:1	Reserved.
0	<b>NmiMask.</b> Read-write. Specifies whether NMI was masked upon entry to SMM. 0=NMI not masked. 1=NMI masked.

### SMMFED8 SMM SVM State

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

Bits	Description														
63:4	Reserved.														
3	<b>HostEflagsIf: host Eflags IF.</b>														
2:0	<b>SvmState.</b> <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>SMM entered from a non-guest state.</td> </tr> <tr> <td>001b</td> <td>Reserved.</td> </tr> <tr> <td>010b</td> <td>SMM entered from a guest state.</td> </tr> <tr> <td>101b-011b</td> <td>Reserved.</td> </tr> <tr> <td>110b</td> <td>SMM entered from a guest state with nested paging enabled.</td> </tr> <tr> <td>111b</td> <td>Reserved.</td> </tr> </tbody> </table>	Bits	Definition	000b	SMM entered from a non-guest state.	001b	Reserved.	010b	SMM entered from a guest state.	101b-011b	Reserved.	110b	SMM entered from a guest state with nested paging enabled.	111b	Reserved.
Bits	Definition														
000b	SMM entered from a non-guest state.														
001b	Reserved.														
010b	SMM entered from a guest state.														
101b-011b	Reserved.														
110b	SMM entered from a guest state with nested paging enabled.														
111b	Reserved.														

### SMMFEFC SMM-Revision Identifier

SMM entry state: 0003\_0064h

Bits	Description
31:18	Reserved.
17	<b>BRL</b> . Read-only. Base relocation supported.
16	<b>IOTrap</b> . Read-only. IO trap supported.
15:0	<b>Revision</b> . Read-only.

#### SMMFF00 SMM Base Address (SMM\_BASE)

Bits	Description
31:0	See: <a href="#">MSRC001_0111</a> [SmmBase].

#### 2.4.8.2.6 Exceptions and Interrupts in SMM

When SMM is entered, the core masks INTR, NMI, SMI, INIT, and A20M interrupts. The core 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 core 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 core responds to the DBREQ and STPCLK interrupts, as well as to all exceptions that may be caused by the SMI handler.

#### 2.4.8.2.7 The Protected ASeg and TSeg Areas

These ranges are controlled by [MSRC001\\_0112](#) and [MSRC001\\_0113](#); see those registers for details.

#### 2.4.8.2.8 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](#)[RsmSpCycDis, SmiSpCycDis].

#### 2.4.8.2.9 Locking SMM

The SMM registers ([MSRC001\\_0112](#) and [MSRC001\\_0113](#)) can be locked from being altered by setting [MSRC001\\_0015](#)[SmmLock]. SBIOS must lock the SMM registers after initialization to prevent unexpected changes to these registers.

#### 2.4.8.2.10 Synchronizing SMM Entry (Spring-Boarding)

The BIOS must take special care to ensure that all cores have entered SMM prior to accessing shared IO resources and all core SMI interrupt status bits are synchronized. This generally requires that BIOS waits for all cores to enter SMM.

The following conditions can cause one or more cores to enter SMM without all cores entering SMM:



- More than one IO device in the system is enabled to signal an SMI without hardware synchronization (e.g. using an end of SMI gate).
- A single device may signal multiple SMI messages without hardware synchronization (e.g. using an end of SMI gate).
- An SMI is received while one or more AP cores are in the INIT state. This may occur either during BIOS or secure boot.
- A hardware error prevents a core from entering SMM.

The act of synchronizing cores into SMM is called spring-boarding. Because not all of the above conditions can be avoided, it is recommended that all systems support spring-boarding.

An ACPI-compliant IO hub is required for spring-boarding. Depending on the IO hub design, BIOS may have to set additional end-of-SMI bits to trigger an SMI from within SMM.

The software requirements for the suggested spring-boarding implementation are listed as follows.

- A binary semaphore located in SMRAM, accessible by all cores. For the purpose of this discussion, the semaphore is called CheckSpringBoard. CheckSpringBoard is initialized to zero.
- Two semaphores located in SMRAM, accessible by all cores. For the purpose of this discussion, the semaphores are called NotInSMM and WaitInSMM. NotInSMM and WaitInSMM are initialized to a value equal to the number of cores in the system (NumCPUs).

The following BIOS algorithm describes spring-boarding and is optimized to reduce unnecessary SMI activity. This algorithm must be made part of the SMM instruction sequence for each core in the system.

1. Attempt to obtain ownership of the CheckSpringBoard semaphore with a read-modify-write instruction. If ownership was obtained then do the following, else proceed to step 2:
  - Check all enabled SMI status bits in the IO hub.  
Let Status=enable1&status1 | enable2&status2 | enable3&status3 ... enable n & status n.
  - If (Status==0) then perform the following sub-actions.
    - Trigger an SMI broadcast assertion from the IO hub by writing to the software SMI command port.
    - Resume from SMM with the RSM instruction.

```
//Example:
InLineASM{
    BTS CheckSpringBoard,0; Try to obtain ownership of semaphore
    JC Step_2:
    CALL CheckIOHUB_SMIEVT; proc returns ZF=1 for no events
    JNZ Step_2:
    CALL Do_SpringBoard;Trigger SMI and then RSM
Step_2:
}
```
2. Decrement the NotInSMM variable. Wait for (NotInSMM==0). See Note 1.
3. Execute the core-local event SMI handler. Using a third semaphore (not described here), synchronize core execution at the end of the task. After all cores have executed, proceed to step 4. The following is a brief description of the task for each core:
  - Check all enabled core-local SMI status bits in the core's private or MSR address space. Handle the event if possible, or pass information necessary to handle the event to a mailbox for the BSC to handle.
  - An exclusive mailbox must exist for each core for each core local event.
  - Wait for all cores to complete this task at least once.
4. If the current core executing instructions is not the BSC then jump to step 5. If the core executing instructions is the BSC then jump to the modified main SMI handler task, described below.
  - Check all enabled SMI status bits in the IO hub. Check mailboxes for event status.

- For each event, handle the event and clear the corresponding status bit.
  - Repeat until all enabled SMI status bits are clear and no mailbox events remain.
  - Set NotInSMM=NumCPUs. (Jump to step 5.)
5. Decrement the WaitInSMM variable. Wait for WaitInSMM=0. See Note 2.
  6. Increment the WaitInSMM variable. Wait for WaitInSMM=NumCPUs.
  7. If the current processor core executing instructions is the BSC then reset CheckSpringBoard to zero.
  8. Resume from SMM with the RSM instruction.

## Notes:

1. To support a secure startup by the secure loader the BIOS must provide a timeout escape from the otherwise endless loop. The timeout value should be large enough to account for the latency of all cores entering SMM. The maximum SMM entrance latency is defined by the platform's IO sub-system, not the processor. A value of twice the watchdog timer count is recommended. See [D18F3x44 \[MCA NB Configuration\]](#) for more information on the watchdog time-out value. If a time-out occurs in the wait loop, the BIOS (the last core to decrement NotInSMM) should record the number of cores that have not entered SMM and all cores must fall out of the loop.
2. If a time-out occurs in the wait loop in step 2, the BIOS must not wait for WaitInSMM=0. Instead it must wait for WaitInSMM=(the number of cores recorded in step 2).
3. If BIOS places APs in the INIT state during any part of the boot process when SMIs may be generated, or may generate SMIs before taking all APs out of their initial microcode reset loop (i.e., before [D18F0x1DC\[CpuEn\]](#) is set), then it is recommended that BIOS keep a record of how many APs are in these two states and exclude these cores from the wait loops. SMIs are not recognized by a processor in these states. AMD does not recommend enabling SMI sources prior to bringing all APs out of these states.

### 2.4.9 Secure Virtual Machine Mode (SVM)

Support for SVM mode is indicated by [CPUID Fn8000\\_0001\\_ECX\[SVM\]](#).

#### 2.4.9.1 BIOS support for SVM Disable

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

- Enable AMD Virtualization™.
  - [MSRC001\\_0114\[SvmeDisable\]](#) = 0.
  - [MSRC001\\_0114\[Lock\]](#) = 1.
  - [MSRC001\\_0118\[SvmLockKey\]](#) = 0000\_0000\_0000\_0000h.
- Disable AMD Virtualization™.
  - [MSRC001\\_0114\[SvmeDisable\]](#)=1.
  - [MSRC001\\_0114\[Lock\]](#)=1.
  - [MSRC001\\_0118\[SvmLockKey\]](#) = 0000\_0000\_0000\_0000h.

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

- Disable AMD Virtualization™, with a user supplied key.
  - [MSRC001\\_0114\[SvmeDisable\]](#)=1.
  - [MSRC001\\_0114\[Lock\]](#)=1.
  - [MSRC001\\_0118\[SvmLockKey\]](#) programmed with value supplied by user. This value should be stored in NVRAM.

## 2.4.10 CPUID Instruction

The CPUID instruction provides data about the features supported by the processor. See [3.17 \[CPUID Instruction Registers\]](#).

### 2.4.10.1 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 8](#) provides a summary of ACPI states and power management features and indicates whether they are supported.

**Table 8: 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	2.5.4.1 [NB P-states]
G0/S0/C0: Hardware thermal control (HTC)	Yes	2.10.3.1 [PROCHOT_L and Hardware Thermal Control (HTC)]
G0/S0/Per-core IO-based C-states	Yes	2.5.3.2 [Core C-states] and 2.5.1.3.2 [Low Power Voltages]
G0/S0/C1: Halt	Yes	
G0/S0/CC6: Per-core Power gating	Yes	2.5.3.2 [Core C-states]
G0/S0: CPC-L2 power gating	Yes	2.5.3.2 [Core C-states]
G0/S0/PC6: 0V support (VDD power plane).	Yes	2.5.3.2 [Core C-states] and 2.5.1.3.2 [Low Power Voltages]
G0/S0/Cx: Cache flushing support	Yes	2.5.3.2.3.1 [C-state Probes and Cache Flushing]
G0/S0: Northbridge C-states (DRAM self-refresh, NB clock-gating)	Yes	2.5.4.2 [NB C-states]
G1/S1: Stand By (Powered On Suspend)	No	
G1/S3: Stand By (Suspend to RAM)	Yes	2.5.8.1 [S-states]
G1/S4: Hibernate (Suspend to Disk)	Yes	
G1/S5: 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 1	No	
Serial VID Interface 2	Yes	
Single-plane systems	No	
Number of voltage planes	2	2.5.1 [Processor Power Planes And Voltage Control]
APM: Application Power Management	Yes	2.5.9 [Application Power Management (APM)]

### 2.5.1 Processor Power Planes And Voltage Control

Refer to the *Electrical Data Sheet* for power plane definitions. See [1.2 \[Reference Documents\]](#).

#### 2.5.1.1 Serial VID Interface

The processor includes an interface to control external voltage regulators, called the serial VID interface (SVI). Only SVI2 is supported. The frequency of SVC for SVI2 is controlled by [D18F3xA0\[Svi2HighFreqSel\]](#). See the *AMD Serial VID Interface 2.0 (SVI2) Specification* for additional details.

##### 2.5.1.1.1 SVI2 Features

The processor supports the following SVI2 features:

- Voltage offsets:
  - VDD: [D18F5x12C\[CoreOffsetTrim\]](#).
  - VDDNB: [D18F5x188\[NbOffsetTrim\]](#).
- Load line trim:
  - VDD: [D18F5x12C\[CoreLoadLineTrim\]](#).
  - VDDNB: [D18F5x188\[NbLoadLineTrim\]](#).

### 2.5.1.2 Internal VID Registers and Encodings

All VID register fields within the processor are 8-bits wide. The VID encodings to voltage translation for all VID codes are defined by the SVI mode. See the *AMD Serial VID Interface 2.0 (SVI2) Specification* for additional details.

The boot VID is 0.9 volts.

#### 2.5.1.2.1 MinVid and MaxVid Check

Hardware limits the minimum and maximum VID code that is sent to the voltage regulator. The allowed limits are specified in [D18F5x17C\[MinVid, MaxVid\]](#). 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.

### 2.5.1.3 Low Power Features

#### 2.5.1.3.1 PSIX\_L Bit

The processor can indicate whether or not it's in a low-voltage state via the PSIX\_L bit. This indicator may be used by the voltage regulator to place itself into a more power efficient mode. PSIX\_L is controlled independently for VDD and VDDNB.

- The processor supports the PSIO\_L and the PSII\_L bits in the data fields of the SVI2 command.
  - PSIO\_L: PSIO\_L for VDD and VDDNB is enabled using [D18F3xA0\[PsiVidEn\]](#) and [D18F5x17C\[NbPsi0VidEn\]](#), respectively. Once enabled, the state of PSIO\_L is controlled by [D18F3xA0\[PsiVid\[7:0\]\]](#) and [D18F5x17C\[NbPsi0Vid\]](#). Changes to the state of PSIO\_L can only occur on VID changes.
  - PSII\_L: The PSII\_L bit for VDD and VDDNB is specified by [D18F5x12C\[CorePsi1En\]](#) and [D0F0xBC\\_x3F9EC\[EnableNbPsi1\]](#), respectively. See also [D18F5x188\[NbPsi1\]](#). Changes to the state of PSII\_L can occur at any time.

##### 2.5.1.3.1.1 BIOS Requirements for PSIO\_L

Enabling PSIO\_L for the VDD and VDDNB planes depends on support from the voltage regulator and is therefore system specific. The voltage regulator must be able to supply the current required for the processor to operate at the VID code specified in [D18F3xA0\[PsiVid\[7:0\]\]](#) and [D18F5x17C\[NbPsi0Vid\[7:0\]\]](#). Depending on the regulator used, AMD recommends one of the following methods:

- PSIO\_L disabled:
  - VDD: To set PSIO\_L for the VDD plane, program `D18F3xA0[PsiVidEn]=0`.
  - VDDNB: To set PSIO\_L for the VDDNB plane, program `D18F5x17C[NbPsi0VidEn]=0`.
- PSIO\_L set/clear based on current requirements:
  - VDD: The following algorithm describes how to program PSIO\_L on VDD:

```

PSI_vrm_current = current at which the regulator allows PSIO_L.
previous_voltage = FFh

for (each P-state from P0 to D18F3xDC[HwPstateMaxVal]) {
  pstate_current = ProcIddMax for the current P-state,
                  see 2.5.3.1.7 [Processor-Systemboard Power Delivery Compatibility Check];
  pstate_voltage = MSRC001_00[6B:64][CpuVid] of the current P-state;

  if (current P-state == D18F3xDC[HwPstateMaxVal]) {
    next_pstate_current = 0;
  } else {
    next_pstate_current = ProcIddMax for the next P-state,
                        see 2.5.3.1.7 [Processor-Systemboard Power Delivery Compatibility Check];
  }

  if ((pstate_current <= PSI_vrm_current) &&
      (next_pstate_current <= PSI_vrm_current) &&
      (pstate_voltage != previous_voltage)) {
    Program D18F3xA0[PsiVid] = pstate_voltage;
    Program D18F3xA0[PsiVidEn] = 1;
    break;
  }
  previous_voltage = pstate_voltage;
}

```

- VDDNB: The following algorithm describes how to program PSIO\_L on VDDNB:

NbIddMax = `D18F5x16[C:0][NbIddDiv]` current.

```

PSI_vrm_current = current at which the VDDNB regulator allows PSIO_L.
previous_voltage = FFh

for (each valid NB P-state starting with NBP0) {
  pstate_current = NbIddMax of the current NB P-state;
  pstate_voltage = D18F5x16[C:0][NbVid] of the current NB P-state;

  if (current P-state is the last valid P-state) {
    next_pstate_current = 0;
  } else {
    next_pstate_current = NbIddMax for the next P-state;
  }

  if ((pstate_current <= PSI_vrm_current) &&
      (next_pstate_current <= PSI_vrm_current) &&
      (pstate_voltage != previous_voltage)) {
    Program D18F5x17C[NbPsi0Vid] = pstate_voltage;
    Program D18F5x17C[NbPsi0VidEn] = 1;
    break;
  }
  previous_voltage = pstate_voltage;
}

```

### 2.5.1.3.2 Low Power Voltages

In order to save power, voltages lower than those normally needed for operation may be applied to the VDD power plane while the processor is in a C-state or S-state. The lower voltage is defined as follows:

- PC6Vid: [D18F5x128](#)[PC6Vid] specifies a voltage that does not retain the CPU caches or the core microarchitectural state. PC6Vid does not allow execution and is only applied to the cores. See [2.5.3.2.3.4 \[Package C6 \(PC6\) State\]](#).

### 2.5.1.4 Voltage Transitions

The processor supports dynamic voltage transitions on the VDD and VDDNB planes. These transitions are requested by either hardware or software during state changes such as reset, P-state changes, and C-state changes. In all cases 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, no stepping occurs. See the *AMD Serial VID Interface 2.0 (SVI2) Specification* for additional details.

- If a voltage increase is requested, the processor waits the amount of time specified by [D18F5x12C](#)[WaitVidCompDis] 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 waits the amount of time specified by [D18F5x128](#)[FastSlamTimeDown] before sending any additional voltage change requests to the voltage regulator. For voltage decreases, the processor does not wait any time before beginning frequency changes.

The processor continues code execution during voltage changes when in the C0 state.

#### 2.5.1.4.1 Hardware-Initiated Voltage Transitions

When software requests any of the following 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. [D18F5x128](#)[PC6Vid] specifies a voltage that does not retain the CPU caches or the core microarchitectural state. PC6Vid does not allow execution and is only applied to the cores. See [2.5.3.2.3.4 \[Package C6 \(PC6\) State\]](#).
  - S-state transition. See [2.5.8.1 \[S-states\]](#).
- VDDNB:
  - NB P-state transition. See [2.5.4.1 \[NB P-states\]](#).
  - S-state transition. See [2.5.8.1 \[S-states\]](#).

#### 2.5.1.4.2 Software-Initiated Voltage Transitions

##### 2.5.1.4.2.1 Software-Initiated NB Voltage Transitions

Software can request voltage changes on the VDDNB power plane using the BIOSMC\_MSG\_VDDNB\_REQUEST software interrupt. To make a voltage change request, software uses the sequence described in [2.12.1 \[Software Interrupts\]](#) with Service Index 3Ah.

Software voltage requests are considered by hardware when taking voltage plane dependencies into account (see [2.5.2.2 \[Dependencies Between Subcomponents on VDDNB\]](#)).

## 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 \[Core C-states\]](#)), hardware must take the following frequency and voltage domain dependencies into account when deciding whether to make the requested change:

- Cores within a compute unit share a common frequency and voltage domain.
- Compute units within a processor share a common voltage domain, but have independent frequency domains. The voltage is determined by the highest-performance P-state requested on any core.

As a result, the P-state and C-state change requests have the following results:

- If different compute units request different voltages, the VDD voltage is determined by the highest voltage (lowest VID) requested.
- If the cores within a compute unit request different P-states while in C0, frequency and voltage are determined by the highest-performance P-state requested.
- If both cores request non-C0 states, the behavior is specified by [D18F4x128\[CoreCstatePolicy\]](#).
- If one core within a compute unit requests a CC6 while the other core is in C0, the frequency and voltage of the compute unit is determined by the core in C0.
- If one core within a compute unit requests a CC6 while the other core is in C0, the core could independently enter CC6.

### 2.5.2.2 Dependencies Between Subcomponents on VDDNB

Many subcomponents of the processor 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 [2.5.1.4.1 \[Hardware-Initiated Voltage Transitions\]](#)), or software makes a voltage change request (see [2.5.1.4.2 \[Software-Initiated Voltage Transitions\]](#)), 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.

### 2.5.2.3 BIOS Requirements for Power Plane Initialization

- Ensure the following fields are configured to their BIOS recommendations:
  - [D18F3xA0\[Svi2HighFreqSel\]](#).
  - [D18F3xD8\[VSRampSlamTime\]](#).
- Optionally configure [PSIx\\_L](#). Refer to [2.5.1.3.1 \[PSIx\\_L Bit\]](#) for additional details.

## 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 (P0 through P7), specified in [MSRC001\\_00\[6B:64\]](#). Out of cold reset, the voltage and frequency of the compute units is specified by [MSRC001\\_0071\[StartupPstate\]](#).



Support for dynamic core P-state changes is indicated by more than one enabled selection in [MSRC001\\_00\[6B:64\]\[PstateEn\]](#). At least one enabled P-state (P0) is specified for all processors.

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](#). Software may not request any P-state transitions using the hardware P-state control mechanism until the P-state initialization requirements defined in [2.5.3.1.6 \[BIOS Requirements for Core P-state Initialization and Transitions\]](#) are complete.

The processor supports independently-controllable frequency planes for each compute unit and the NB; and independently-controllable voltage planes. See [2.5.1 \[Processor Power Planes And Voltage Control\]](#) for voltage plane definitions.

The following terms may be applied to each of these planes:

- FID: frequency ID. Specifies the PLL frequency multiplier, relative to the reference clock, for a given domain.
- DID: divisor ID. Specifies the post-PLL power-of-two divisor that can be used to reduce the operating frequency.
- COF: current operating frequency. Specifies the operating frequency as a function of the FID and DID. Refer to [CoreCOF](#) for the CPU COF formula and [NBCOF](#) for the NB COF formula.
- VID: voltage ID. Specifies the voltage level for a given domain. Refer to [2.5.1.2.1 \[MinVid and MaxVid Check\]](#) for encodings.

All FID and DID parameters for software P-states must be programmed to equivalent values for all cores and NBs in the coherent fabric. See [2.5.3.1.1.1 \[Software P-state Numbering\]](#). Refer to [MSRC001\\_00\[6B:64\]](#) and [D18F5x16\[C:0\]](#) for further details on programming requirements.

### 2.5.3.1.1 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.

#### 2.5.3.1.1.1 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 9: Software P-state Naming**

D18F4x15C[NumBoost-States]=1		D18F4x15C[NumBoost-States]=3	
P-state Name	Corresponding MSR Address	P-state Name	Corresponding 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.

#### 2.5.3.1.1.2 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.

#### 2.5.3.1.2 Core P-state Control

Core P-states are dynamically controlled by software and are exposed through ACPI objects (refer to [2.5.3.1.8.3 \[ACPI Processor P-state Objects\]](#)). Software requests a core P-state change by writing a 3 bit index corresponding to the desired P-state number to [MSRC001\\_0062\[PstateCmd\]](#) 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 APM (i.e. writes 000b to [MSRC001\\_0062\[PstateCmd\]](#)), hardware dynamically places the core into the highest-performance P-state possible as determined by APM. See [2.5.9 \[Application Power Management \(APM\)\]](#).

Hardware sequences the frequency and voltage changes necessary to complete a P-state transition as specified by [2.5.3.1.5 \[Core P-state Transition Behavior\]](#) 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.

**Table 10: Software P-state Control**

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

Hardware controls the VID for each voltage domain according to the highest requirement of the frequency domain(s) on each plane. For example, the VID for a 4 compute unit dual-plane system must be maintained at the highest level required for all 4 frequency domains. The number of frequency domains in a voltage domain is package/platform specific. Refer to 2.5.3.1.5 [Core P-state Transition Behavior] for details on hardware P-state voltage control. 2.5.2.3 [BIOS Requirements for Power Plane Initialization] specifies the processor initialization requirements for voltage plane control.

### 2.5.3.1.3 Core P-state Visibility

MSRC001\_0063[CurPstate] reflects the current non-boosted P-state number for each compute unit. For example, if MSRC001\_0063[CurPstate]=010b on compute unit 1, then compute unit 1 is in the P2 state. If a compute unit is in a boosted P-state, MSRC001\_0063[CurPstate] reads back as 0.

The voltage on a compute unit may not correspond to the VID code specified by the current P-state of the compute unit due to voltage plane dependencies. See 2.5.2 [Frequency and Voltage Domain Dependencies]. If a compute unit is in the P0 state (i.e. if MSRC001\_0063[CurPstate]=0), the frequency of the compute unit could be the frequency specified by P0 or any boosted P-state. To determine the frequency of a compute unit, see 2.5.3.3 [Effective Frequency].

### 2.5.3.1.4 Core P-state Limits

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

- HTC. See D18F3x64[HtcPstateLimit].
- Software. See D18F3x68[SwPstateLimit].
- Core Performance Boost. See 2.5.9 [Application Power Management (APM)].
- PROCHOT\_L assertion. See 2.10.3.1 [PROCHOT\_L and Hardware Thermal Control (HTC)].
- SMU. See D18F4x13C[SmuPstateLimit].

P-state limits are applied to all cores on the processor. The current P-state limit is provided in MSRC001\_0061[CurPstateLimit]. Changes to the MSRC001\_0061[CurPstateLimit] can be programmed to trigger interrupts through D18F3x64[PslApicLoEn, PslApicHiEn]. In addition, the maximum P-state value, regardless of the source, is limited as specified in MSRC001\_0061[PstateMaxVal].

### 2.5.3.1.5 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 compute unit 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.
- When the processor initiates a VID change that increases voltage for a voltage domain, no new voltage or frequency changes occur until [D18F3xD8\[VSRampSlamTime\]](#) has expired, regardless of whether any new requests are received. When the processor initiates a VID change that decreases voltage for a voltage domain, new voltage or frequency changes are allowed to occur immediately.
  - This is true regardless of whether the frequency or voltages changes occur as a result of P-state or C-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 PWROK deassertion. If multiple P-state changes are requested concurrently, the PSSM may group the associated VID changes separately from the associated COF changes.
- Behavior during RESET\_L assertions:
  - All cores are transitioned to C0.
  - VDD VID is transitioned to HWP0 VID. See [MSRC001\\_0064\[CpuVid\]](#).
  - If there is no P-state transition activity, then the compute units and NB remain in the current P-state.
  - If a RESET\_L assertion interrupts a P-state transition, then the COF remains in it's current state at the time RESET\_L is asserted (either the value of the old or the new P-state). BIOS is required to transition to valid COF and VID settings after a warm reset according to the sequence defined in [2.5.3.1.8 \[BIOS COF and VID Requirements After Warm Reset\]](#).
  - After a warm reset [MSRC001\\_0063 \[P-state Status\]](#) is consistent with [MSRC001\\_0071\[CurPstate\]](#). [MSRC001\\_0062 \[P-state Control\]](#) may not be consistent with [MSRC001\\_0071\[CurPstate\]](#).
- The OS controls the P-state through [MSRC001\\_0062 \[P-state Control\]](#), independent of P-state limits described in [D18F3x64\[HtcPstateLimit\]](#), [D18F3x68\[SwPstateLimit\]](#). P-state limits interact with OS-directed P-state transitions as follows:
  - Of all the active P-state limits, the one that represents the lowest-performance P-state number, at any given time, is treated as an upper limit on performance.
  - As the limit becomes active or inactive, or if it changes, the P-state for each compute unit is placed in either the last OS-requested P-state or the new limit P-state, whichever is a lower performance P-state number.
    - If the resulting P-state number exceeds [MSRC001\\_0061\[PstateMaxVal\]](#), regardless of whether it is a limit or OS-requested, then the PstateMaxVal is used instead.

### 2.5.3.1.6 BIOS Requirements for Core P-state Initialization and Transitions

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 [2.5.2.3 \[BIOS Requirements for Power Plane Initialization\]](#).
3. Ensure the following fields are configured to their BIOS recommendations:
  - [D18F3xA0\[PIILockTime\]](#).
  - [D18F3xD4\[PowerStepUp, PowerStepDown\]](#).
4. Transition all cores to the minimum performance P-state using the algorithm detailed in [2.5.3.1.8.2 \[Core Minimum P-state Transition Sequence After Warm Reset\]](#).
5. Complete the [2.5.4.1.2.1 \[NB P-state COF and VID Synchronization After Warm Reset\]](#). All cores on a processor must be in the minimum performance P-state prior to executing this sequence.

6. Complete the 2.5.3.1.7 [Processor-Systemboard Power Delivery Compatibility Check].
7. Perform the following steps in any order:
  - A. Enable 2.5.9 [Application Power Management (APM)] as follows:
    - Ensure the following fields are configured to their BIOS recommendations:
      - D18F4x110[CSampleTimer].
      - D18F4x15C[ApmMasterEn].
      - D18F5xE0[RunAvgRange].
    - See your AMD representative for details on how to enable the GPU aspects of 2.5.9 [Application Power Management (APM)].
    - If D18F4x15C[NumBoostStates]! = 0, program D18F4x15C[BoostSrc]=1.
  - B. Transition all cores to the maximum performance P-state by writing 0 to MSRC001\_0062[PstateCmd].
  - C. Create ACPI objects if necessary:
    - Determine the valid set of P-states as indicated by MSRC001\_00[6B:64][PstateEn].
    - If P-states are not supported, as indicated by only one enabled selection in MSRC001\_00[6B:64][PstateEn], then BIOS must not generate ACPI-defined P-state objects described in 2.5.3.1.8.3 [ACPI Processor P-state Objects]. Otherwise, the ACPI objects should be generated to enable P-state support.
  - D. Configure the COF and VID for each processor appropriately based on the sequence described in 2.5.4.1.2 [BIOS NB P-state Configuration].
8. Configure PSIX\_L. Refer to 2.5.1.3.1 [PSIX\_L Bit] for additional details.

### 2.5.3.1.7 Processor-Systemboard Power Delivery Compatibility Check

BIOS must disable processor 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 power plane(s) that contain the core(s). Refer to 2.5.1 [Processor Power Planes And Voltage Control] for power plane definitions and configuration information. BIOS can optionally notify the user if P-states are detected that exceed the systemboard power delivery capability. Modifications to MSRC001\_00[6B:64] [P-state [7:0]] must be applied equally to all cores on the same node. This check does not ensure 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.

$$\text{ProcIddMax} = \text{MSRC001\_00[6B:64][IddValue]} \text{ current} * 1/10^{\text{MSRC001\_00[6B:64][IddDiv]}} * (\text{D18F5x84[CmpCap]}+1)$$

The power delivery check should be applied starting with hardware P0 and continue with increasing P-state indexes (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. All processor P-states with higher indexes are defined to be lower power and performance, and are therefore compatible with the systemboard.

Example:

- MSRC001\_0065[IddValue] = 32d
- MSRC001\_0065[IddDiv] = 0d
- D18F5x84[CmpCap] = 1d
- ProcIddMax = 32 \* 1 \* 2 = 64A per plane

The systemboard must be able to supply >= 64A for the unified core power plane in order to support P1 for this

processor. If the systemboard current delivery capability is < 64A per plane then BIOS must set `MSRC001_0065[PstateEn]=0` for all cores on this node, and continue by checking P2 in the same fashion.

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

If at least one P-state is disabled by performing the power delivery compatibility check and at least one P-state remains enabled, then BIOS must perform the following steps:

1. 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.
2. Copy the contents of the enabled P-state MSR (`MSRC001_00[6B:64]`) to the highest performance P-state locations. E.g. if P0 and P1 are disabled by the power delivery compatibility check and P2 - P4 remain enabled, then the contents of P2 - P4 should be copied to P0 - P2 and P3 and P4 should be disabled (`PstateEn=0`). This step uses software P-state numbering. See 2.5.3.1.1.1 [Software P-state Numbering].
3. 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.
4. If a subset of boosted P-states are disabled, then copy the contents of the P-state MSR pointed to by the highest performance boosted P-state that is enabled to the P-state MSRs pointed to by the boosted P-states that are disabled.
5. If all boosted P-states are disabled, then program `D18F4x15C[BoostSrc]=0`.
6. Adjust the following P-state parameters affected by the P-state MSR copy by subtracting the number of software 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]`
  - `D18F3x68[SwPstateLimit]`
  - `D18F3xDC[HwPstateMaxVal]`

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

1. If `MSRC001_0063[CurPstate] != MSRC001_0061[PstateMaxVal]`, then write `MSRC001_0061[PstateMaxVal]` to `MSRC001_0062[PstateCmd]` and wait for `MSRC001_0063[CurPstate]` to reflect the new value.
2. If `MSRC001_0061[PstateMaxVal] != 000b` copy the contents of the P-state MSR pointed to by `MSRC001_0061[PstateMaxVal]` 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. This step uses software P-state numbering. See 2.5.3.1.1.1 [Software P-state Numbering].
3. Adjust the following fields to 000b.
  - `D18F3x64[HtcPstateLimit]`
  - `D18F3x68[SwPstateLimit]`
  - `D18F3xDC[HwPstateMaxVal]`
4. Program `D18F4x15C[BoostSrc]=0`.

### 2.5.3.1.8 BIOS COF and VID Requirements After Warm Reset

Warm reset is asynchronous and can interrupt P-state transitions leaving the processor in a VID state that does not correspond to `MSRC001_0063[CurPstate]` on any core. The processor frequency after warm reset

corresponds to `MSRC001_0063[CurPstate]`. See 2.5.3.1.5 [Core P-state Transition Behavior] for P-state transition behavior when `RESET_L` is asserted. BIOS is required to transition the processor to valid COF and VID settings corresponding to an enabled P-state following warm reset. The cores may be transitioned to either the maximum or minimum P-state COF and VID settings using the sequences defined in 2.5.3.1.8.1 [Core Maximum P-state Transition Sequence After Warm Reset] and 2.5.3.1.8.2 [Core Minimum P-state Transition Sequence After Warm Reset]. Transitioning to the minimum P-state after warm reset is recommended to prevent undesired system behavior if a warm reset occurs before the 2.5.3.1.7 [Processor-Systemboard Power Delivery Compatibility Check] is complete. BIOS is not required to manipulate NB COF and VID settings following warm reset if the warm reset was issued by BIOS to update `D18F5x16[C:0][NbFid]`.

#### 2.5.3.1.8.1 Core Maximum P-state Transition Sequence After Warm Reset

1. If `MSRC001_0071[CurPstate] = D18F3xDC[HwPstateMaxVal]`, then skip step 3 for that core.
2. Write `MSRC001_0061[PstateMaxVal]` to `MSRC001_0062[PstateCmd]` on all cores in the processor.
3. Wait for `MSRC001_0071[CurCpuFid, CurCpuDid] = [CpuFid, CpuDid]` from `MSRC001_00[6B:64]` indexed by `D18F3xDC[HwPstateMaxVal]`.
4. Wait for `MSRC001_0071[CurCpuVid] = [CurCpuVid]` from `MSRC001_00[6B:64]` indexed by `D18F3xDC[HwPstateMaxVal]`.
5. All previous steps must be completed on all cores prior to continuing the sequence since a compute unit transitions to the highest performance P-state requested on either core.
6. Write 0 to `MSRC001_0062[PstateCmd]` on all cores in the processor.
7. Wait for `MSRC001_0071[CurCpuFid, CurCpuDid] = [CpuFid, CpuDid]` from `MSRC001_00[6B:64]` indexed by `MSRC001_0071[CurPstateLimit]`.
8. If `MSRC001_0071[CurPstateLimit] != D18F3xDC[HwPstateMaxVal]`, wait for `MSRC001_0071[CurCpuVid] = [CpuVid]` from `MSRC001_00[6B:64]` indexed by `MSRC001_0071[CurPstateLimit]`.
9. Wait for `MSRC001_0063[CurPstate] = MSRC001_0061[CurPstateLimit]`.

#### 2.5.3.1.8.2 Core Minimum P-state Transition Sequence After Warm Reset

1. If `MSRC001_0071[CurPstate] = MSRC001_0071[CurPstateLimit]`, then skip step 3 for that core.
2. Write 0 to `MSRC001_0062[PstateCmd]` on all cores in the processor.
3. Wait for `MSRC001_0071[CurCpuFid, CurCpuDid] = [CpuFid, CpuDid]` from `MSRC001_00[6B:64]` indexed by `MSRC001_0071[CurPstateLimit]`.
4. Write `MSRC001_0061[PstateMaxVal]` to `MSRC001_0062[PstateCmd]` on all cores in the processor.
5. Wait for `MSRC001_0071[CurCpuFid, CurCpuDid] = [CpuFid, CpuDid]` from `MSRC001_00[6B:64]` indexed by `D18F3xDC[HwPstateMaxVal]`.
6. If `MSRC001_0071[CurPstateLimit] != MSRC001_0071[CurPstate]`, wait for `MSRC001_0071[CurCpuVid] = [CpuVid]` from `MSRC001_00[6B:64]` indexed by `D18F3xDC[HwPstateMaxVal]`.
7. Wait for `MSRC001_0063[CurPstate] = MSRC001_0062[PstateCmd]`.

#### 2.5.3.1.8.3 ACPI Processor P-state Objects

Processor performance control is implemented through the `_PCT`, `_PSS` and `_PSD` objects in ACPI 2.0 and later revisions. The presence of these objects indicates to the OS that the platform and processor are capable of supporting multiple performance states. Processor performance states are not supported with ACPI 1.0b. BIOS must provide the `_PCT`, `_PSS`, and `_PSD` objects, and define other ACPI parameters to support operating systems that provide native support for processor P-state transitions.

The following rules apply to BIOS generated ACPI objects in multi-core systems. Refer to the appropriate

ACPI specification 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-consumptions parameters as P0 on core 1, 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.

### 2.5.3.1.8.3.1 **\_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

### 2.5.3.1.8.3.2 **\_PSS (Performance Supported States)**

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

BIOS loops through each of `MSRC001_00[6B:64]` applying the following formulas to create the fields for the `_PSS` object for for each valid P-state (see `MSRC001_00[6B:64][PstateEn]`). BIOS skips over any P-state MSRs that specify boost P-states (see `D18F4x15C[NumBoostStates]`).

- CoreFreq (MHz) = Calculated using the formula for CoreCOF.
- Power (mW) = `MSRC001_00[6B:64][CpuVid]` voltage \* `MSRC001_00[6B:64][IddValue]` current \* 1000.
- TransitionLatency (us) and BusMasterLatency (us):
  - If `MSRC001_00[6B:64][CpuFid]` is the same for all enabled P-states (see `MSRC001_00[6B:64][PstateEn]`) and all boosted P-states:
    - $\text{TransitionLatency} = \text{BusMasterLatency} = (15 \text{ steps} * \text{D18F3xD4}[\text{PowerStepDown}] \text{ time} * 1000 \text{ us/ns}) + (15 \text{ steps} * \text{D18F3xD4}[\text{PowerStepUp}] \text{ time} * 1000 \text{ us/ns})$
  - Else if `MSRC001_00[6B:64][CpuFid]` is different for any enabled (see `MSRC001_00[6B:64][PstateEn]`) or boost P-states:
    - $\text{TransitionLatency} = \text{BusMasterLatency} = (15 \text{ steps} * \text{D18F3xD4}[\text{PowerStepDown}] \text{ time} * 1000 \text{ us/ns}) + \text{D18F3xA0}[\text{PllLockTime}] \text{ time} + (15 \text{ steps} * \text{D18F3xD4}[\text{PowerStepUp}] \text{ time} * 1000 \text{ us/ns})$
  - Example:
    - `MSRC001_00[6B:64][CpuFid]` is not the same for all P-states
    - `D18F3xD4[PowerStepDown] = D18F3xD4[PowerStepUp] = 8h (50 ns/step)`
    - `D18F3xA0[PllLockTime] = 001b (2 us)`
    - $\text{TransitionLatency} = \text{BusMasterLatency} = (15 \text{ steps} * 50 \text{ ns/step} / 1000 \text{ us/ns}) + 2\text{us} + (15 \text{ steps} * 50 \text{ ns/step} / 1000 \text{ us/ns}) = 3.5 \text{ us (round up to 4 us)}$
- Control/Status:
  - The highest performance non-boosted P-state must have the `_PSS` control and status fields programmed to 0.
  - Any lower performance non-boosted P-states must have the `_PSS` control and status fields programmed



in ascending order.

#### 2.5.3.1.8.3.3 **\_PPC (Performance Present Capabilities)**

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

#### 2.5.3.1.8.3.4 **\_PSD (P-state Dependency)**

AMD recommends the ACPI 3.0 `_PSD` object be generated for each core as follows to cause the cores to transition between P-states independently:

- `NumberOfEntries = 5.`
- `Revision = 0.`
- `Domain = CPUID Fn0000_0001_EBX[LocalApicId[7:2]].`
- `CoordType = FEh. (HW_ALL)`
- `NumProcessors = 4.`

#### 2.5.3.1.8.4 **Fixed ACPI Description Table (FADT) Entries**

Declare the following FADT entries:

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

#### 2.5.3.1.8.5 **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 **Core 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. When coming out of warm and cold reset, the processor is transitioned to the C0 state.

#### 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-states is not direct. AMD specified C-states are referred to as IO-based C-states. Up to three IO-based C-states are supported, IO-based C-state 0, 1, and 2. The IO-based C-state index corresponds to the offset added to `MSRC001_0073[CstateAddr]` to initiate a C-state request. See [2.5.3.2.2 \[C-state Request Interface\]](#). The actions taken by the processor when entering a low-power C-state 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 [2.5.3.2.6 \[ACPI Processor C-state Objects\]](#)). C-states can be requested on a per-core basis. Software requests a C-state change in one of two ways:

- Reading from an IO address: The IO address must be the address specified by `MSRC001_0073[CstateAddr]`

plus an offset of 0 through 7. The processor always returns 0 for this IO read. Offsets 2 through 7 result in an offset of 2.

- Executing the HLT instruction. This is equivalent to reading from the IO address specified by [D18F4x128\[HaltCstateIndex\]](#).

When software requests a C-state transition, hardware evaluates any frequency and voltage domain dependencies and determines which C-state actions to execute. See [2.5.2 \[Frequency and Voltage Domain Dependencies\]](#) and [2.5.3.2.3 \[C-state Actions\]](#).

### 2.5.3.2.3 C-state Actions

A core takes one of several different possible actions based upon a C-state change request from software. The C-state action fields are defined in [D18F4x11\[C:8\]](#).

#### 2.5.3.2.3.1 C-state Probes and Cache Flushing

If probes occur after a core enters a non-C0 state, and the caches are not flushed by hardware, the core clock may be ramped back up to the C0 frequency to service the probes, as specified by [D18F4x118/D18F4x11C\[CpuPrbEn\]](#).

If a core enters a non-C0 state and cache flush is enabled (see [D18F3xDC\[CacheFlushOnHaltCtl\]](#) and [D18F4x118/D18F4x11C\[CacheFlushEn\]](#)), a timer counts down for a programmable period of time as specified by [D18F3xDC\[CacheFlushOnHaltTmr\]](#) or [D18F4x118/D18F4x11C\[CacheFlushTmrSel\]](#). When the timer expires, the core flushes its L1 cache to L2 cache and the core clocks are ramped down to a divisor specified by [D18F3xDC\[CacheFlushOnHaltCtl\]](#). The timer is reset if the core exits the C-state for any reason. See [2.5.3.2.4.2 \[Cache Flush On Halt Saturation Counter\]](#).

Once a core flushes its caches probes are no longer sent to that core. This improves probing performance for cores that are in C0.

#### 2.5.3.2.3.2 Core C1 (CC1) State

When a core enters the CC1 state, its clock ramps down to the frequency specified by [D18F4x118/D18F4x11C\[ClkDivisorCstAct\]](#).

#### 2.5.3.2.3.3 Core C6 (CC6) State

A core can gate off power to its internal logic when it enters any non-C0 state. This power gated state is known as CC6. In order to enter CC6, hardware first enters CC1 then checks [D18F4x118/D18F4x11C\[PwrGateEnCstAct\]](#). Power gating reduces the amount of power consumed by the core. VDD voltage is not reduced when a core is in CC6. The following sequence occurs when a core enters the CC6 state:

1. If [MSRC001\\_0071\[CurPstate\]](#) < [D18F3xA8\[PopDownPstate\]](#), transition the core P-state to [D18F3xA8\[PopDownPstate\]](#).
2. Internal core state is saved to L1 cache .
3. L1 cache is flushed to L2 cache. See [2.5.3.2.3.1 \[C-state Probes and Cache Flushing\]](#).
4. Power is removed from the core and the core PLL/voltage regulator is powered down as specified by [D18F5x128\[CC6PwrDwnRegEn\]](#).

All of the following must be true in order for a core to be placed into CC6:

- [D18F4x118/D18F4x11C\[CacheFlushEn\]](#)=1 for the corresponding C-state action field.
- [D18F4x118/D18F4x11C\[CacheFlushTmrSel\]](#) != 11b for the corresponding C-state action field.

- D18F4x118/D18F4x11C[PwrGateEnCstAct]=1 for the corresponding C-state action field.
- D18F2x118[CC6SaveEn]=1.
- D18F2x118[LockDramCfg]=1 .
- The CC6 storage area in DRAM is configured. See 2.9.11 [DRAM CC6/PC6 Storage].

The events which cause a core to exit the CC6 state are specified in 2.5.3.2.5 [Exiting C-states].

If a warm reset occurs while a core is in CC6, all MCA registers in the core shown in Table 65 are cleared to 0. See 2.14.1 [Machine Check Architecture].

The time required to enter and exit CC6 is directly proportional to the core P-state frequency. Slower core frequencies require longer entry and exit times. Latency issues may occur with core P-state frequencies less than 800MHz.

#### 2.5.3.2.3.4 Package C6 (PC6) State

When all cores enter a non-C0 state, VDD can be reduced to a non-operational voltage that does not retain core state. This state is known as PC6 and reduces the amount of static and dynamic power consumed by all cores.

The following actions are taken by hardware prior to PC6 entry:

1. If MSRC001\_0071[CurPstate] < D18F3xA8[PopDownPstate], transition the core P-state to D18F3xA8[PopDownPstate].
2. For all cores not in CC6, internal core state is saved to L1 cache .
3. For all cores not in CC6, L1 cache is flushed to L2 cache. See 2.5.3.2.3.1 [C-state Probes and Cache Flushing].
4. VDD is transitioned to the VID specified by D18F5x128[PC6Vid].
5. If the core PLLs are not powered down during CC6 entry (see 2.5.3.2.3.3 [Core C6 (CC6) State]), then they are powered down as specified by D18F5x128[PC6PwrDwnRegEn].

All of the following must be true on all cores in order for a package to be placed into PC6:

- D18F4x118/D18F4x11C[CacheFlushEn]=1 for the corresponding C-state action field
- D18F4x118/D18F4x11C[CacheFlushTmrSel] != 11b for the corresponding C-state action field.
- D18F4x118/D18F4x11C[PwrOffEnCstAct]=1 for the corresponding C-state action field.
- D18F2x118[CC6SaveEn]=1.
- D18F2x118[LockDramCfg]=1.

#### 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 FCH Messaging

The FCH can be notified when the processor transitions package C-states. See the following:

- D18F4x128[CstateMsgDis].
- D18F5x178[CstateFusionDis].

##### 2.5.3.2.4.2 Cache Flush On Halt Saturation Counter

A cache flush success monitor tracks the success rate of cache flush timer expirations relative to the core

exiting a C-state. Based on the success rate, caches may be flushed immediately without waiting for the cache flush timer to expire. See [D18F4x128](#)[CacheFlushSucMonThreshold]. When the core resumes normal execution, the caches refill as normal.

#### 2.5.3.2.5 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 at the interrupt service routine.
- Entry via IO read, EFLAGS[IF]==0: The interrupt wakes the core and the core begins execution at the instruction after the IN instruction that was used to enter the non-C0 C-state.
- Entry via IO read, EFLAGS[IF]==1: The interrupt wakes the core and the core begins execution at the interrupt service routine.

#### 2.5.3.2.6 ACPI Processor C-state Objects

Processor power control is implemented through the `_CST` object in ACPI 2.0 and later revisions. The presence of the `_CST` object 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. See [2.5.3.2.6.1 \[\\_CST\]](#).

The `_CST` object is not supported with ACPI 1.0b. BIOS should provide FADT entries to support operating systems that are not compatible with ACPI 2.0 and later revisions. See [2.5.3.2.6.2 \[\\_CRS\]](#).

##### 2.5.3.2.6.1 `_CST`

The `_CST` object should be generated for each core as follows:

- Count = 1.
- Register = [MSRC001\\_0073](#)[CstateAddr] + 1.
- Type = 2.
- Latency = 400.
- Power = 0.

##### 2.5.3.2.6.2 `_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.

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

Declare the following FADT entries:

- P\_LVL2\_LAT = 100.
- P\_LVL3\_LAT = 1001.

- `FLAGS.PROC_C1 = 1.`
- `FLAGS.P_LVL2_UP = 1.`

Declare the following `P_BLK` entries:

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

BIOS must declare the `PSTATE_CNT` entry as 00h.

### 2.5.3.2.7 BIOS Requirements for Initialization

1. Initialize `MSRC001_0073[CstateAddr]` with an available IO address. See 2.5.3.2.6.2 [`_CRS`].
2. Initialize `D18F4x11[C:8]`.
3. Generate ACPI objects as described in 2.5.3.2.6 [`ACPI Processor C-state Objects`].

### 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 limited by:

- HTC
- `D18F3x68[SwPstateLimit]`
- SBI
- CPB

The following procedure calculates effective frequency using `MSR0000_00E7 [Max Performance Frequency Clock Count (MPERF)]` and `MSR0000_00E8 [Actual Performance Frequency Clock Count (APERF)]`:

1. At some point in time, write 0 to both MSR.
2. At some later point in time, read both MSR.
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 at most 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.
- `MSR0000_00E7` and `MSR0000_00E8` are invalid if an overflow occurs.

## 2.5.4 NB Power Management

### 2.5.4.1 NB P-states

The processor supports up to four NB P-states (NBP0 through NBP3), specified in `D18F5x16[C:0]`. Each NB

P-state consists of the following:

- Enable: [D18F5x16\[C:0\]\[NbPstateEn\]](#).
- NCLK frequency: [D18F5x16\[C:0\]\[NbFid, NbDid\]](#).
- VDDNB voltage: [D18F5x16\[C:0\]\[NbVid\]](#).
- Memory P-state: [D18F5x16\[C:0\]\[MemPstate\]](#). See [2.5.7.1 \[Memory P-states\]](#).

Out of cold reset, the NB P-state is specified by [D18F5x174\[StartupNbPstate\]](#) and [D18F3xA0\[CofVidProg\]](#). The current NB P-state is specified by [D18F5x174\[CurNbFid, CurNbDid, CurNbVid\]](#).

Although four NB P-states are defined, only two NB P-states are used at any given time, specified by [D18F5x170\[NbPstateHi, NbPstateLo\]](#).

#### 2.5.4.1.1 NB P-state Transitions

Hardware selects whether to use the high or low NB P-state based on several criteria as follows:

- Core P-state:
  - [MSRC001\\_00\[6B:64\]\[NbPstate\]](#).
  - [D18F5x170\[NbPstateThreshold\]](#).
- GPU activity:
  - The GPU driver selects levels of GPU activity that force the NB P-state to either the high or low state or allow either NB P-state.
- Hysteresis timer:
  - [D18F5x170\[NbPstateHiRes, NbPstateLoRes\]](#).
- The following configuration registers:
  - [D18F5x170\[SwNbPstateLoDis, NbPstateDisOnP0\]](#).
  - [MSRC001\\_0071\[NbPstateDis\]](#).

Once the NB determines that an NB P-state transition is necessary, the NB executes the following sequence:

1. If transitioning from the low NB P-state to the high NB P-state, transition VDDNB voltage.
2. If the GPU is enabled as specified by [D18F5x178\[SwGfxDis\]](#), wait for display buffer to fill.
3. Quiesce all active cores.
4. If the internal GPU is enabled as specified by [D18F5x178\[SwGfxDis\]](#), wait for display buffer to fill.
5. Stop memory traffic and place DRAM into self-refresh.
6. Transition NCLK frequency.
7. Update NB P-state specific DRAM settings within hardware, see [D18F2x210\\_dct\[0\]\\_nbp\[3:0\]](#).
8. Take DRAM out of self-refresh and allow memory traffic.
9. Wake up cores.
10. If transitioning from the high NB P-state to the low NB P-state, transition VDDNB voltage.

#### 2.5.4.1.2 BIOS NB P-state Configuration

##### 2.5.4.1.2.1 NB P-state COF and VID Synchronization After Warm Reset

BIOS performs the following sequence on one core. This is done after any warm reset and before [2.9.7 \[DCT/DRAM Initialization and Resume\]](#).

1. Temp1=[D18F5x170\[SwNbPstateLoDis\]](#).
2. Temp2=[D18F5x170\[NbPstateDisOnP0\]](#).
3. Temp3=[D18F5x170\[NbPstateThreshold\]](#).

4. Temp4=D18F5x170[NbPstateGnbSlowDis].
5. If MSRC001\_0070[NbPstate]=0, go to step 6. If MSRC001\_0070[NbPstate]=1, go to step 11.
6. Write 1 to D18F5x170[NbPstateGnbSlowDis].
7. Write 0 to D18F5x170[SwNbPstateLoDis, NbPstateDisOnP0, NbPstateThreshold].
8. Wait for D18F5x174[CurNbPstate] = D18F5x170[NbPstateLo] and D18F5x174[CurNbFid, CurNbDid]=[NbFid, NbDid] from D18F5x16[C:0] indexed by D18F5x170[NbPstateLo].
9. Set D18F5x170[SwNbPstateLoDis]=1.
10. Wait for D18F5x174[CurNbPstate] = D18F5x170[NbPstateHi] and D18F5x174[CurNbFid, CurNbDid]=[NbFid, NbDid] from D18F5x16[C:0] indexed by D18F5x170[NbPstateHi]. Go to step 15.
11. Write 1 to D18F5x170[SwNbPstateLoDis].
12. Wait for D18F5x174[CurNbPstate] = D18F5x170[NbPstateHi] and D18F5x174[CurNbFid, CurNbDid]=[NbFid, NbDid] from D18F5x16[C:0] indexed by D18F5x170[NbPstateHi].
13. Write 0 to D18F5x170[SwNbPstateLoDis, NbPstateDisOnP0, NbPstateThreshold].
14. Wait for D18F5x174[CurNbPstate] = D18F5x170[NbPstateLo] and D18F5x174[CurNbFid, CurNbDid]=[NbFid, NbDid] from D18F5x16[C:0] indexed by D18F5x170[NbPstateLo].
15. Set D18F5x170[SwNbPstateLoDis]=Temp1, D18F5x170[NbPstateDisOnP0]=Temp2, and D18F5x170[NbPstateThreshold]=Temp3, and D18F5x170[NbPstateGnbSlowDis]=Temp4.

#### 2.5.4.1.2.2 NB P-state Transitions

During boot when D18F5x174[NbPstateDis]=0, BIOS forces the processor to the desired NB P-states using the following steps:

1. Save the values in D18F5x170 for later restoration to unforce the NB P-state.
2. Set the desired NB P-state pointers, D18F5x170[NbPstateHi, NbPstateLo].
3. Transition to the desired state as follows:
  - In order to transition to D18F5x170[NbPstateHi], program D18F5x170 as follows:
    - SwNbPstateLoDis = 1.
    - Wait for D18F5x174[CurNbPstate] to equal NbPstateHi.
  - In order to transition to D18F5x170[NbPstateLo], program D18F5x170 as follows:
    - SwNbPstateLoDis = NbPstateDisOnP0 = NbPstateThreshold = 0.
    - Wait for D18F5x174[CurNbPstate] to equal NbPstateLo.

BIOS performs the following to release the NB P-state force:

4. Release the NB P-state force by restoring initial D18F5x170 values.
  - Restore the initial D18F5x170[SwNbPstateLoDis, NbPstateDisOnP0, NbPstateLo] values.
  - Restore the initial D18F5x170[NbPstateThreshold, NbPstateHi] values.

#### 2.5.4.1.2.3 NB P-state Configuration for Runtime

Please see your AMD representative for details.

#### 2.5.4.2 NB C-states

NB C-states are package-level actions that occur only when all cores enter a non-C0 state (see 2.5.3.2 [Core C-states]). The NB C-state actions are:

- DRAM self-refresh (see 2.5.7.2 [DRAM Self-Refresh]):
  - Enable bit: D18F4x118/D18F4x11C[SelfRefr].
  - Entry requirements:
    - No outstanding GPU traffic or traffic from a link.

- Exit conditions (any of the following must be true):
  - The local APIC timer expires. See 2.4.8.1 [Local APIC].
  - New GPU traffic or traffic from a link.
  - A P-state limit update (see 2.5.3.1.4 [Core P-state Limits]) causes the most restrictive P-state limit to become a higher number than the current P-state for any core in CC1.
- NB clock gating:
  - Enable bit: D18F4x118/D18F4x11C[NbClkGate].
  - Entry requirements:
    - No outstanding GPU traffic or traffic from a link.
  - Exit conditions (any of the following must be true):
    - The local APIC timer expires. See 2.4.8.1 [Local APIC].
    - New GPU traffic or traffic from a link.
    - A P-state limit update (see 2.5.3.1.4 [Core P-state Limits]) causes the most restrictive P-state limit to become a higher number than the current P-state for any core in CC1.

When entering NB C-states, the actions are taken in the following order:

1. DRAM self-refresh.
2. NB clock gating.

When exiting NB C-states, the actions are taken in the following order:

1. NB clock gating.
2. DRAM self-refresh.

### 2.5.5 Bandwidth Requirements

- The frequency relationship of  $(\text{core COF} / \text{NB COF}) \leq 6$  &  $(\text{core COF} / \text{NB COF}) \geq 1$  must be maintained for all supported P-state combinations. E.g., a core P0 COF of 4.0 GHz could not be combined with a NB P0 COF of 0.6 GHz; the NB P0 COF would have to be 0.8 GHz or greater; if the NB P0 COF is 1.2 GHz, then the NB P1 COF of 0.6 GHz may only be supported if the corresponding core P-state specify a COF of 3.0 GHz or less.
- All core P-states are required to be defined such that  $(\text{NB COF}/\text{core COF}) \leq 32$ , for all NB/core P-state combinations. E.g., if the NB COF is 4.8 GHz then the core COF must be no less than 150 MHz.
- All core P-states must be defined such that  $\text{CoreCOF} \geq 600$  MHz.
- $\text{NBCOF} \geq \text{MEMCLK}$  frequency.
- $400\text{MHz} \leq \text{NBCOF} \leq 1600\text{MHz}$ .
- $\text{NBCOF} \leq 2 * \text{MEMCLK}$  frequency.

### 2.5.6 GPU and Root Complex Power Management

#### 2.5.6.1 Dynamic Power Management (DPM)

The processor supports dynamic GPU 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 DPM, higher numbered states represent higher performance and lower numbered states represent lower performance.

##### 2.5.6.1.1 Activity Monitors

The processor contains activity monitors which track the usage level of different processor subcomponents. A



binary signal from each subcomponent is used to determine whether that subcomponent is busy. On each clock cycle, the activity monitor samples the signal from each unmasked subcomponent.

See [D0F0xBC\\_xC020\\_0110](#) for LCLK DPM activity monitor.

The output of the activity monitor is then used to determine whether the DPM state should be changed.

### 2.5.6.1.2 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.

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

### 2.5.6.1.3 LCLK DPM

LCLK 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: [D0F0xBC\\_x3FD\[8C:00:step14\]\[StateValid\]](#).
- Voltage change hysteresis threshold: [D0F0xBC\\_x3FD\[8C:00:step14\]\[LowVoltageReqThreshold\]](#).
- Divisor: [D0F0xBC\\_x3FD\[8C:00:step14\]\[LclkDivider\]](#).
- VID: [D0F0xBC\\_x3FD\[8C:00:step14\]\[VID\]](#).
  - See [2.5.2.2 \[Dependencies Between Subcomponents on VDDNB\]](#).
- State change hysteresis thresholds: [D0F0xBC\\_x3FD\[94:08:step14\]\[HysteresisUp, HysteresisDown\]](#).
- Activity thresholds: [D0F0xBC\\_x3FD\[9C:10:step14\]\[ActivityThreshold\]](#).
- Residency counter: See [D0F0xBC\\_x3FD\[94:08:step14\]\[ResidencyCounter\]](#).

LCLK DPM is enabled by setting [D0F0xBC\\_x3FDC8\[LclkDpmEn\]](#). LCLK DPM voltage changes are enabled using [D0F0xBC\\_x3FDC8\[VoltageChgEn\]](#). When LCLK DPM is first enabled, the DPM state is transitioned to [D0F0xBC\\_x3FDC8\[LclkDpmBootState\]](#).

### 2.5.6.2 GPU and Root Complex Power Gating

Several subcomponents of the GPU and root complex can be power gated when not in use.

- **GPU:** GPU power gating is initialized and enabled by software (see [GpuEnabled](#) and [D0F0x7C\[ForceIntGfxDisable\]](#)). Once initialized and enabled, the GPU is power gated by hardware when inactive and is ungated by hardware when needed. When internal GPU is disabled by BIOS, BIOS is responsible for power gating the GPU.
- **UVD:** UVD is power gated by software when not in use and is ungated by software when needed. UVD's internal state is not saved and UVD goes through an internal reset when power is restored.
- **GMC:** GMC power gating is initialized and enabled by software. Once initialized and enabled, GMC is power gated by hardware when inactive and is ungated by hardware when needed. GMC's state is saved internally. If the internal GPU is disabled, either by hardware (fusing) or by software, software is responsible for power gating GMC.

- **VCE:** VCE is power gated by software when not in use and is ungated by software when needed. VCE's internal state is not saved and VCE goes through an internal reset when power is restored.
- **DCE:** DCE is power gated by software when there is no display connected. DCE's internal state is not saved and DCE goes through an internal reset when power is restored.
- **PCIe®:** The Gfx link core can be power gated when it is not in use. In addition, the individual phys on the TX and RX sides of each link can be power gated when the links are not connected. During POST and runtime, several software components inform the SMU whether the Gfx link core or the link phys are in use. Hardware dynamically power gates or ungates the Gfx link core and the link phys as needed.

## 2.5.7 DRAM Power Management

### 2.5.7.1 Memory P-states

The processor supports up to 2 memory P-states, M0 and M1. Each memory P-state consists of the following:

- MEMCLK frequency
- A set of frequency dependent DRAM timing and configuration registers

See 2.9 [DRAM Controllers (DCTs)] for DRAM technology specific information and requirements.

All valid memory P-states are associated with a specific NB P-state, as specified by [D18F5x16\[C:0\]\[MemPstate\]](#). When hardware transitionsto a new NB P-state, the memory P-state is transitioned to that specified by the new NB P-state.

Out of cold reset the current memory P-state is M0. The P-state value specified by [D18F5x16\[C:0\]\[MemPstate\]](#) of the NB P-state indexed by [D18F5x174\[StartupNbPstate\]](#) is invalid. Support for dynamic memory P-state changes is indicated by [D18F3xE8\[MemPstateCap\]=1](#) and one or more [D18F5x16\[C:0\]\[MemPstate\]=1](#); otherwise M0 is used by hardware for configuration purposes.

During boot, and if [D18F5x170\[MemPstateDis\]=0](#), the BIOS can disable memory P-states using the following steps:

1. Program [D18F5x170\[MemPstateDis\]=1](#).
2. Program [D18F5x16\[C:0\]\[MemPstate\]=0](#).

### 2.5.7.2 DRAM Self-Refresh

DRAM is placed into self-refresh on S3 entry (see 2.5.8.1.1 [ACPI Suspend to RAM State (S3)]).

In addition to S3, DRAM is placed into self-refresh in S0 in the following two scenarios:

- NB P-state transitions (see 2.5.4.1 [NB P-states]).
- NB C-states (see 2.5.4.2 [NB C-states]).

The following requirements must be met before hardware places DRAM into self-refresh:

- No pending traffic.
- One of the following is true:
  - The GPU is idle and the internal display buffer is full.
  - The internal GPU is disabled.

Once the above requirements are met, hardware places DRAM into self-refresh.

Early self-refresh occurs when DRAMs are placed in self-refresh before expiration of the cache flush timer. See [D18F4x118/D18F4x11C\[SelfRefrEarly\]](#) and [D18F5x128\[SelfRefrEarlyDis\]](#). If early self-refresh is enabled, the DRAMs are taken out of self-refresh to perform the flush operation when the cache flush timer expires and then placed back into self-refresh.

The following are events that cause DRAM to transition out of self-refresh:

- Core transitioning to C0.
- Incoming request from any link or the GMC
- P-state limit update, only in the case when all cores are not in the power gated (CC6) state.

To save additional power, hardware always tristates MEMCLK when entering self-refresh.

### 2.5.7.3 Stutter Mode

DRAM is most commonly placed in self-refresh due to stutter mode when the internal GPU is in use. 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.

#### 2.5.7.3.1 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.

### 2.5.7.4 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 (see [D18F2xA4\[CmdThrottleMode\]](#)) until DRAM training is complete.

See [2.9.12 \[DRAM On DIMM Thermal Management and Power Capping\]](#).

## 2.5.8 System Power Management

### 2.5.8.1 S-states

S-states are ACPI defined sleep states. S0 is the operational state. All other S-states are low-power states in which various voltage rails in the system may or may not be powered. See the ACPI specification for descriptions of each S-state. The only other S-state supported is S5.

### 2.5.8.1.1 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, software is responsible for transitioning the processor to Memory Pstate0. See 2.5.7.1 [Memory P-states].

During S3 entry, system memory enters self-refresh mode (see 2.5.7.2 [DRAM Self-Refresh]). Software is responsible for bringing memory out of self-refresh mode when resuming from S3. To bring memory out of self-refresh mode. See 2.9.7 [DCT/DRAM Initialization and Resume].

Many of the systemboard power planes for the processor are powered down during S3. Refer to the Electrical Data Sheet 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.

## 2.5.9 Application Power Management (APM)

Application Power Management (APM) allows the processor to deterministically provide maximum performance while remaining within the specified power delivery and removal envelope. APM 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 APM hardware to reduce power consumption. APM 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 APM.

### 2.5.9.1 Core Performance Boost (CPB)

These P-states are referred to as boosted P-states.

- Support for APM is specified by [CPUID Fn8000\\_0007\\_EDX\[CPB\]](#).
- APM is enabled if all of the following conditions are true:
  - [MSRC001\\_0015\[CpbDis\]](#) = 0 for all cores.
  - [D18F4x15C\[ApmMasterEn\]](#) = 1.
  - [D18F4x15C\[BoostSrc\]](#) = 01b.
  - [D18F4x15C\[NumBoostStates\]](#) != 0.
- APM can be dynamically enabled and disabled through [MSRC001\\_0015\[CpbDis\]](#). If core performance boost (CPB) is disabled, a P-state limit is applied. The P-state limit restricts cores 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.
- Two levels of boosted P-states are supported. Compute units can be placed in the first level of boosted P-states if the processor power consumption remains within the TDP limit. The second level of boosted P-

states is C-state Boost. See [2.5.9.1.1 \[C-state Boost\]](#).

- All boosted P-states are always higher performance than non-boosted P-states.
- To ensure proper operation, boosted P-states should be hidden from the operating system. BIOS should not provide ACPI \_PSS entries for boosted P-states. See [2.5.3.1.8.3.2 \[\\_PSS \(Performance Supported States\)\]](#).
- The lowest-performance P-state CPB limits the processor to is the highest-performance non-boosted P-state.

#### **2.5.9.1.1 C-state Boost**

C-state Boost can only be achieved if a subset of compute units are in CC6 and the processor power consumption remains within the TDP limit. See [D18F4x16C\[CstateCnt, CstateBoost\]](#).

#### **2.5.9.2 TDP Limiting**

TDP limiting is a mechanism for capping the power consumption of the processor through a TDP limit.

#### **2.5.9.3 Bidirectional Application Power Management (BAPM)**

Bidirectional Application Power Management (BAPM) is an algorithm to enable fine grained power transfers between the core and GPU.

## 2.6 Performance Monitoring

The processor includes support for two methods of monitoring processor performance:

- [2.6.1 \[Performance Monitor Counters\]](#).
- [2.6.2 \[Instruction Based Sampling \(IBS\)\]](#).

### 2.6.1 Performance Monitor Counters

The following types of performance counters are supported:

- [2.6.1.1 \[Core Performance Monitor Counters\]](#), consisting of one set located in each core of each compute unit.
- [2.6.1.2 \[L2I Performance Monitor Counters\]](#), consisting of one set located in each compute unit.
- [2.6.1.3 \[NB Performance Monitor Counters\]](#), consisting of one set located in each node.

The accuracy of the performance counters is not ensured. 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.

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.

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.

In addition to the RDMSR instruction, the performance counter registers can be read using a special read performance-monitoring counter instruction, RDPMC.

#### 2.6.1.1 Core Performance Monitor Counters

The core performance monitor counters are used by software to count specific non L2 events that occur in a core of the compute unit. Each core of each compute unit provides four 48-bit performance counters. Unless otherwise specified, the events count only the activity of the core, not activity caused by the other core of the compute unit.

[MSRC001\\_00\[03:00\]](#) specify the events to be monitored and how they are monitored. All of the events are specified in [3.22 \[Core Performance Counter Events\]](#). [MSRC001\\_00\[07:04\]](#) are the counters.

All of the events are specified in [3.22 \[Core Performance Counter Events\]](#).

#### 2.6.1.2 L2I Performance Monitor Counters

The L2I performance monitor counters are used by software to count specific L2 events that occur in a core of the compute unit. Unless otherwise specified using the thread masks, the events count the activity of all cores. Each compute unit provides four 48-bit performance counters.

[MSRC001\\_023\[6,4,2,0\] \[L2I Performance Event Select \(L2I\\_PERF\\_CTL\[3:0\]\)\]](#) specify the events to be mon-

itored and how they are monitored. [MSRC001\\_023\[7,5,3,1\]](#) [[L2I Performance Event Counter \(L2I\\_PERF\\_CTR\[3:0\]\)](#)] are the counters. Support for [MSRC001\\_023\[7:0\]](#) is indicated by [CPUID Fn8000\\_0001\\_ECX\[PerfCtrExtL2I\]](#).

All of the events are specified in [3.23 \[L2I Performance Counter Events\]](#).

All L2I performance monitor events can be counted on all counters.

All L2I performance events are one event per clock.

### 2.6.1.3 NB Performance Monitor Counters

The NB performance monitor counters are used by software to count specific events that occur in the NB. Each node provides four 48-bit performance counters. Since the northbridge performance counter register are shared by all cores on a node, monitoring of northbridge events should only be performed by one core on a node.

[MSRC001\\_024\[6,4,2,0\]](#) [[Northbridge Performance Event Select \(NB\\_PERF\\_CTL\[3:0\]\)](#)] and [MSRC001\\_024\[7,5,3,1\]](#) [[Northbridge Performance Event Counter \(NB\\_PERF\\_CTR\[3:0\]\)](#)] specify the events to be monitored and how they are monitored. Support for [MSRC001\\_024\[7:0\]](#) is indicated by [CPUID Fn8000\\_0001\\_ECX\[PerfCtrExtNB\]](#).

All of the events are specified in [3.24 \[NB Performance Counter Events\]](#).

All NB performance monitor events can be counted on all counters.

All NB performance events are one event per clock.

NB performance counters do not support APIC interrupt capability.

### 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 [MSRC001\\_103A \[IBS Control\]](#). 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 by [MSRC001\\_1030 \[IBS Fetch Control \(IbsFetchCtl\)\]](#); and instruction execution performance controlled by [MSRC001\\_1033 \[IBS Execution Control \(IbsOpCtl\)\]](#). 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 independent 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 for the tagged instruction fetch:

- If the instruction fetch completed or was aborted. See [MSRC001\\_1030](#).
- The number of clock cycles spent on the instruction fetch. See [MSRC001\\_1030](#).
- If the instruction fetch hit or missed the IC, hit/missed in the L1 and L2 TLBs, and page size. See [MSRC001\\_1030](#).
- The linear address, physical address associated with the fetch. See [MSRC001\\_1031](#), [MSRC001\\_1032](#).

Instruction execution performance is profiled by tagging one micro-op associated with an instruction. Instructions that decode to more than one micro-op return different performance data depending upon which micro-op associated with the instruction is tagged. These micro-ops are associated with the RIP of the next instruction to retire. The following performance information is returned for the tagged micro-op:

- Branch and execution status for micro-ops. See [MSRC001\\_1035](#).
- Branch target address for branch micro-ops. See [MSRC001\\_103B](#).
- The logical address associated with the micro-op. See [MSRC001\\_1034](#).
- The linear and physical address associated with a load or store micro-op. See [MSRC001\\_1038](#), [MSRC001\\_1039](#).
- The data cache access status associated with the micro-op: DC hit/miss, DC miss latency, TLB hit/miss, TLB page size. See [MSRC001\\_1037](#).
- The number clocks from when the micro-op was tagged until the micro-op retires. See [MSRC001\\_1035](#).
- The number clocks from when the micro-op completes execution until the micro-op retires. See [MSRC001\\_1035](#).
- Source information for DRAM and MMIO. See [MSRC001\\_1036](#).

## 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. Processor configuration space is accessed through bus 0, devices 18h to 1Fh, where device 18h corresponds to node 0 and device 1Fh corresponds to node 7. See [2.7.3 \[Processor Configuration Space\]](#).

Configuration space is accessed by the processor through two methods:

- IO-space configuration: IO instructions to addresses CF8h and CFCh.
  - Enabled through [IOCF8\[ConfigEn\]](#), which allows access to BCS.
  - Access to ECS enabled through [MSRC001\\_001F\[EnableCf8ExtCfg\]](#).
  - Use of IO-space configuration can be programmed to generate GP faults through [MSRC001\\_0015\[IoCf-gGpFault\]](#).
  - SMI trapping for these accesses is specified by [MSRC001\\_0054 \[IO Trap Control \(SMI\\_ON\\_IO\\_TRAP\\_CTL\\_STS\)\]](#) and [MSRC001\\_00\[53:50\] \[IO Trap \(SMI\\_ON\\_IO\\_TRAP\\_\[3:0\]\)\]](#).
- MMIO configuration: configuration space is a region of memory space.
  - The base address and size of this range is specified by [MSRC001\\_0058 \[MMIO Configuration Base Address\]](#). 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:
    - $\text{Address}[31:0] = \{0\text{h}, \text{bus}[7:0], \text{device}[4:0], \text{function}[2:0], \text{offset}[11:0]\}$ .

The BIOS may use either configuration space access mechanism during boot. Before booting the OS, BIOS must disable IO access to ECS, enable MMIO configuration and build an ACPI defined MCFG table. BIOS ACPI code must use MMIO to access configuration space.

Per the link specification, BCS accesses utilize link addresses starting at FD\_FE00\_0000h and ECS accesses utilize link addresses starting at FE\_0000\_0000h.

### 2.7.1 MMIO Configuration Coding Requirements

MMIO configuration space accesses must use 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 used 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 core 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. 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 Northbridge (NB)

Each node includes a single northbridge that provides the interface to the local core(s), the interface to system memory, and the interface to 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 2.4.6 [System Address Map].

### 2.8.1 NB Architecture

Major NB blocks are: System Request Interface (SRI), Memory Controller (MCT), DRAM Controllers (DCTs), and crossbar (XBAR). SRI interfaces with the core(s). MCT maintains cache coherency and interfaces with the DCTs; MCT maintains a queue of incoming requests called MCQ. XBAR is a switch that routes packets between SRI, MCT, and the link.

The MCT operates on physical addresses. Before passing transactions to the DCTs, the MCT converts physical addresses into *normalized* addresses that correspond to the values programmed into `D18F2x[5C:40]_dct[0]` [DRAM CS Base Address]. Normalized addresses include only address bits within the DCTs' range.

## 2.8.2 NB Routing

### 2.8.2.1 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
4. Configuration space.

#### 2.8.2.1.1 DRAM and MMIO Memory Space

Memory space transactions provide the NB with the physical address, cacheability type, access type, and DRAM/MMIO destination type as specified in section 2.4.6.1.2 [Determining The Access Destination for Core Accesses].

Memory-space transactions are handled by the NB as follows:

- IO-device accesses are compared against:
  - If the access matches `D18F1x[2CC:2A0,1CC:180,BC:80]` [MMIO Base/Limit], then the transaction is routed to the root complex;
  - Else, if the access matches `D18F1x[17C:140,7C:40]` [DRAM Base/Limit], 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 `D18F1x[17C:140,7C:40]` [DRAM Base/Limit], 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 the VGA-compatible MMIO address space and `D18F1xF4[VE]=1` then `D18F1xF4` describes how the access is routed and controlled;
    - Else, If the access matches `D18F1x[2CC:2A0,1CC:180,BC:80]` [MMIO Base/Limit], then the transaction is routed to the root complex;
    - Else, the access is routed to the UMI.

#### 2.8.2.1.2 IO Space

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

- If the access matches `D18F1x[DC:C0]` [IO-Space Base/Limit], then the transaction is routed to the root complex;
- Else, If the access matches the VGA-compatible IO address space and `D18F1xF4[VE]=1` then `D18F1xF4` describes how the access is routed and controlled.
- Else, the access is routed to the UMI.

#### 2.8.2.1.3 Configuration Space

Configuration-space transactions from IO links are master aborted. Configuration-space transactions from cores are routed as follows:

- If the access matches `D18F1x[1DC:1D0,EC:E0]` [Configuration Map], then the transaction is routed to the specified link;
- Else, the access is routed to link that contains compatibility (subtractive) address space.

### 2.8.2.1.3.1 Recommended Buffer Count Settings Overview

When changing from the recommended settings, see the register programming requirements in the definition of each register. Some chipsets may further optimize these settings for their platform. If values other than the recommended settings are used, see the register requirements in the definition of each register. [Table 11](#) defines commonly used terms for the following tables.

**Table 11: ONION Link Definitions**

Term	Definition
<b>LinkGanged</b>	Ganged = 0.

### 2.8.3 Memory Scrubbers

The processor includes memory scrubbers specified in [MSRC001\\_10A0](#), [D18F3x58](#), [D18F3x5C](#), and [D18F3x60](#). The scrubbers ensure that all cachelines in memory within or connected to the processor are periodically read and, if correctable errors are discovered, they are corrected.

For recommendations on scrub rates, see [2.14.1.8 \[Scrub Rate Considerations\]](#).

The scrub rate is specified as the time between successive scrub events. A scrub event occurs when a line of memory is checked for errors; the amount of memory that is checked varies based on the memory block (see field descriptions).

The time required to fully scrub the memory of a node is determined as:

- $\text{Time} = ((\text{memory size in bytes})/64) * (\text{Scrub Rate})$ .
- E.g. If a node contains 1GB of system memory and  $\text{DramScrub}=5.24$  ms, then all of the system memory of the node is scrubbed about once every 23 hours.

## 2.9 DRAM Controllers (DCTs)

The processor includes one DRAM controller (DCT). 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. BIOS reads [D18F5x84\[DctEn\]](#) to determine the DCT to DDR channel mapping as follows:

- DCT0 controls channel A.

The DCTs operate on normalized addresses corresponding to the values programmed into [D18F2x\[5C:40\]\\_dct\[0\]](#). Normalized addresses only include address bits within a DCT's range. The physical to normalized address translation varies based on various Northbridge settings. See [2.9.10 \[Memory Hoisting\]](#) and [2.9.11 \[DRAM CC6/PC6 Storage\]](#).

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

- All DIMMs connected to a node are required to operate at the same MEMCLK frequency, regardless of the channel. All DCTs of a node must be programmed to the same frequency.
- Mixing of DIMM types within a system is not supported.
- RDIMM, LRDIMM, and stacked DRAM type DIMMs are not supported.
- Mixing of ECC and non-ECC DIMMs within a system is not supported

### 2.9.1 Common DCT Definitions

**Table 12: DCT Definitions**

Term	Definition								
<b>AutoSelfRefresh</b>	SPDByte[31][2] of the DIMM being configured.								
<b>DataMaskMbType</b>	Motherboard type for processor Data Mask pins. <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>No connect</td> </tr> <tr> <td>01b</td> <td>Pins are routed per DM rules</td> </tr> <tr> <td>10b</td> <td>Pins are routed per DQS rules</td> </tr> </tbody> </table>	Bits	Description	00b	No connect	01b	Pins are routed per DM rules	10b	Pins are routed per DQS rules
Bits	Description								
00b	No connect								
01b	Pins are routed per DM rules								
10b	Pins are routed per DQS rules								
<b>DdrRate</b>	The DDR data rate (MT/s) as specified by <a href="#">D18F2x94_dct[0][MemClkFreq]</a> and <a href="#">D18F2x2E0_dct[0][M1MemClkFreq]</a> .								
<b>DeviceWidth</b>	SPDByte[7][2:0] of the DIMM being configured.								
<b>DIMM</b>	The DIMM being configured								
<b>DIMM0</b>	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.								
<b>DIMM1</b>									
<b>DimmsPopulated</b>	The number of DIMMs populated per channel plus rows of <a href="#">Solder-down DRAM</a> devices.								
<b>DR</b>	Dual Rank								
<b>DramCapacity</b>	SPDByte[4][3:0] of the DIMM being configured.								
<b>ExtendedTemperature-Range</b>	SPDByte[31][0] of the DIMM being configured.								
<b>MRS</b>	JEDEC defined DRAM Mode Register Set.								
<b>NP</b>	No DIMM populated								

Table 12: DCT Definitions

Term	Definition
<b>NumDimmSlots</b>	The number of motherboard DIMM slots per channel plus rows of <b>Solder-down DRAM</b> devices
<b>NumRanks</b>	SPDByte[7][5:3] of the DIMM being configured, or the number of ranks soldered down.
<b>Rank</b>	The rank being configured
<b>RankMap</b>	SPDByte[63][0] of the DIMM being configured.
<b>RowAddrBits</b>	SPDByte[5][5:3] of the DIMM being configured.
<b>Solder-down DRAM</b>	DRAM devices soldered directly to the motherboard.
<b>SODIMM</b>	DCT is configured for SODIMM if (D18F2x90_dct[0][UnbuffDimm]==1) and SODIMMs are populated.
<b>SPD</b>	Serial Presence Detect. In the case of DRAMs soldered on the platform, this refers to a virtual representation of the DRAM vendors' data sheets.
<b>SR</b>	Single Rank
<b>UDIMM</b>	DCT is configured for UDIMM if (D18F2x90_dct[0][UnbuffDimm]==1) and UDIMMs are populated.
<b>VDDIO</b>	DDR VDDIO in V.

### 2.9.2 DCT Frequency Support

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 13: DDR3 UDIMM Maximum Frequency Support with 6-layer Motherboard Design FT3 Package

Num-DimmSlots	Dimms Populated	DRAM		Frequency <sup>1</sup> (MT/s)		
		SR	DR	1.5V	1.35V	1.25V
1	1	1	-	1600	-	-
		-	1	1600	-	-
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.

Table 14: DDR3 **UDIMM** Maximum Frequency Support with 6-layer Motherboard Design FT3 Package Microserver

Num-DimmSlots	Dimms Populated	DRAM		Frequency <sup>1</sup> (MT/s)		
		SR	DR	1.5V	1.35V	1.25V
1	1	1	-	1600	1600	-
		-	1	1600	1600	-
2	1	1	-	1600	1600	-
		-	1	1600	1600	-
	2	2	-	1600	1600	-
		1	1	1333	1333	-
-	2	1333	1333	-		

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

Table 15: DDR3 **SODIMM** Maximum Frequency Support with 6-layer Motherboard Design FT3, FS1b Package

Num-DimmSlots	Dimms Populated	DRAM		Frequency <sup>1</sup> (MT/s)		
		SR	DR	1.5V	1.35V	1.25V
1	1	1	-	1600	1600	1333
		-	1	1600	1600	1333
2	1	1	-	1600	1600	1333
		-	1	1600	1600	1333
	2	2	-	1600	1600	1333
		1	1	1600	1333	1333
-	2	1600	1333	1333		

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

Table 16: DDR3 **UDIMM** Maximum Frequency Support with 4-layer Motherboard Design FT3 Package

Num-DimmSlots	Dimms Populated	DRAM		Frequency <sup>1</sup> (MT/s)		
		SR	DR	1.5V	1.35V	1.25V
1	1	1	-	1600	1600	-
		-	1	1600	1600	-
2	1	1	-	1600	1600	-
		-	1	1600	1600	-
	2	2	-	1600	1600	-
		1	1	1333	1333	-
-	2	1333	1333	-		

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

Table 17: DDR3 SODIMM Maximum Frequency Support with 4-layer Motherboard Design FT3, FS1b Package

Num-DimmS-lots	Dimms Populated	DRAM		Frequency <sup>1</sup> (MT/s)		
		SR	DR	1.5V	1.35V	1.25V
1	1	1	-	1600	1333	1066
		-	1	1600	1333	1066
2	1	1	-	1333	1333	1066
		-	1	1333	1333	1066
	2	2	-	1333	1333	1066
		1	1	1333	1333	1066
		-	2	1333	1333	1066

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

Table 18: DDR3 SODIMM plus Solder-down DRAM Maximum Frequency with 6-layer Motherboard FT3 Package

Num-DimmS-lots	Dimms Populated	DRAM		Frequency <sup>1</sup> (MT/s)		
		SR	DR	1.5V	1.35V	1.25V
2	1	1	-	1333	1333	1066
		-	1	1333	1333	1066
	2	2	-	1333	1333	1066
		1	1	1333	1066	1066
		-	2	1333	1066	1066

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

Table 19: DDR3 SODIMM plus Solder-down DRAM Maximum Frequency with 4-layer Motherboard FT3 Package

Num-DimmS-lots	Dimms Populated	DRAM		Frequency <sup>1</sup> (MT/s)		
		SR	DR	1.5V	1.35V	1.25V
2	1	1	-	1066	1066	1066
		-	1	1066	1066	1066
	2	2	-	1066	1066	1066
		1	1	1066	1066	1066
		-	2	1066	1066	1066

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

Table 20: DDR3 Solder-down DRAM Maximum Frequency Support FT3 Package

Num-DimmSlots	Dimms Populated	DRAM		Frequency <sup>1</sup> (MT/s)		
		SR	DR	1.5V	1.35V	1.25V
1	1	1	-	1600	1600	1333
		-	1	1600	1600	1333

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

Table 21: DDR3 UDIMM Maximum Frequency Support FS1b Package

Num-DimmSlots	Dimms Populated	DRAM		Frequency <sup>1</sup> (MT/s)		
		SR	DR	1.5V	1.35V	1.25V
1	1	1	-	1600	1600	1600
		-	1	1600	1600	1333
2	1	1	-	1600	1600	1600
		-	1	1600	1600	1333
	2	2	-	1600	1333	1333
		1	1	1600	1333	1333
		-	2	1600	1333	1333

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

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 22: DDR3 Population Support

NumDimmSlots	DIMM0	DIMM1
1	SR/DR	N/A
2	NP	SR/DR
	SR/DR	SR/DR

### 2.9.3 DCT Configuration Registers

There are multiple types of DCT configuration registers:

- Registers for which there is one instance. E.g. `D18F2xA4` or `D18F2x78_dct[0]`.
- Registers for which there is one instance per NbPstate use `D18F1x10C[NbPsSel]` for software accesses.
  - `D18F2x210_dct[0]_nbp[3:0]` refers to all instances of the D18F2x210 register.
  - `D18F2x210_dct[0]_nbp[1]` refers to the register for Nb P-state 1 of any or all DCTs.
- Registers for which there is one instance per memory P-state use `D18F1x10C[MemPsSel]` for software accesses. The syntax for this register type is described by example as follows:
  - `D18F2x2E8_dct[0]_mp[1:0]` refers to all instances of the D18F2x2E8 register.
  - `D18F2x2E8_dct[0]_mp[1]` refers to the register for memory P-state 1 of either or both DCTs.



## 2.9.4 DDR Pad to Processor Pin Mapping

The relationship of pad drivers to processor pins varies by package as shown in the following table.

**Table 23: Package pin mapping**

Pad	Pin <sup>1</sup>
	FT3
MEMCLK0_H[0]	MA_CLK_H[0]
MEMCLK0_H[1]	MA_CLK_H[1]
MEMCLK0_H[2]	MA_CLK_H[2]
MEMCLK0_H[3]	MA_CLK_H[3]
MEMCLK0_H[4]	NC
MEMCLK0_H[5]	NC
MEMCS0_L[0]	MA0_CS_L[0]
MEMCS0_L[1]	MA0_CS_L[1]
MEMCS0_L[2]	MA1_CS_L[0]
MEMCS0_L[3]	MA1_CS_L[1]
MEMCS0_L[4]	NC
MEMCS0_L[5]	NC
MEMCS0_L[6]	NC
MEMCS0_L[7]	NC
MEMODT0[0]	MA0_ODT[0]
MEMODT0[1]	MA0_ODT[1]
MEMODT0[2]	MA1_ODT[0]
MEMODT0[3]	MA1_ODT[1]
MEMCKE0[0]	MA_CKE[0]
MEMCKE0[1]	MA_CKE[1]
MEMCKE0[2]	MA_CKE[2]
MEMCKE0[3]	MA_CKE[3]
1. For differential pins, only positive polarity pins are shown; negative polarity pins have corresponding mapping and are controlled by the same CSR field. NC = Not connected. BIOS should tri-state or disable the pad for maximum power savings.	

### 2.9.4.1 DDR Chip to Pad Mapping

The relationship of chip to pad drivers is shown in the following table. BIOS should disable or power down unused chips for maximum power savings.

**Table 24: Pad (from chiplet) pin mapping**

Chip	Group	Pad <sup>1</sup>
Clk chip 0	0	MEMCLK0_H[1:0]
Clk chip 1	0	MEMCLK0_H[3:2]
Cmd/Addr 0	0	MEMCS0_L[7,5,3,1]
	1	MEMODT0[3:0]
	2	MEMCS0_L[6,4,2,0]
Cmd/Addr 1	0	MEMRAS0_L
		MEMCAS0_L
		MEMWE0_L
		MEMADD0[13]
	1	MEMADD0[10,0]
		MEMBANK0[1:0]
2	MEMPAR0	
Address 2	0	MEMCKE0[3:0]
	1	MEMADD0[0,1,8,9]
	2	MEMADD0[2,3,10,11]
	3	MEMADD0[4,5,12,13]
	4	MEMADD0[6,7,14,15]
Data <sup>2</sup>	0	MEMDATA[5,4,1,0]
	1	MEMDQS_H[0], MEMDQSDM[0]
	2	MEMDATA[7,6,3,2]
<p>1. For differential pads, only positive polarity pads are shown; negative polarity pads have corresponding mapping and are controlled by the same chip. Only channel A map is shown. For multi-channel products the other channels are similar.</p> <p>2. Only Data chip 0 is shown. Remaining data chips are repeated with sequential DQ/DQS/DM pin numbers.</p>		

### 2.9.5 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 [D18F2x7C\\_dct\[0\]\[EnDramInit\]](#) is set to 1. See [2.3.3 \[Using L2 Cache as General Storage During Boot\]](#).

### 2.9.6 DRAM Data Burst Mapping

DRAM requests are mapped to data bursts on the DDR bus in the following order:

- When [D18F2x110\[DctDatIntLv\]](#) = 0, a 64 B request is mapped to each of the eight sequential data beats as QW0, QW1...QW7.
- When [D18F2x110\[DctDatIntLv\]](#) = 1, the order of cache data to QW on the bus is the same except that even and odd bits are interleaved on the DRAM bus as follows:

- For every 8 bytes in the cache line, even bits map to QW0, QW2, QW4, and QW6 on the DRAM bus.
- For every 8 bytes in the cache line, odd bits map to QW1, QW3, QW5, and QW7 on the DRAM bus.

### 2.9.7 DCT/DRAM Initialization and Resume

DRAM initialization involves several steps in order to configure the DRAM controllers and the DRAM, and to tune the DRAM channel for optimal performance.

DRAM resume requires several steps to configure the DCTs to properly resume from the S3 state. The following sequence describes the steps needed after a reset for initialization or resume:

1. Configure the DDR supply voltage regulator. See 2.9.7.1.
2. Force NB P-state to NBP0.
3. Force register accesses to M0. See 2.9.7.3 for further requirements used in steps below.
4. DDR phy initialization. See 2.9.7.4.
5. 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 2.9.7.5.
    - b. Program Non-SPD configuration. See 2.9.7.6.
    - c. Program DCT training specific configuration. See 2.9.7.7.
    - d. Program the remaining DCT registers not covered by an explicit sequence dependency.
    - e. DRAM device initialization. See 2.9.7.8.
  - 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. See 2.9.7.5 and 2.9.7.6 for a review of registers.
    - b. Program D18F2x90\_dct[0][ExitSelfRef] = 1.
    - c. Restore the trained delayed values (found during the initial boot in step 6 below) from nonvolatile storage.
    - d. Continue at step 8.
6. DRAM data training.
  - A. Write levelization training. See 2.9.7.9.1.
  - B. DQS receiver enable training. See 2.9.7.9.2 and 2.9.7.9.3.
  - C. DQS position training. See 2.9.7.9.4.
7. NB P-state specific training. For each NB P-state to D18F5x170[NbPstateMaxVal]:
  - A. Force the NB P-state. See 2.9.7.2.
  - B. MaxRdLatency training. See 2.9.7.9.5.1.
8. Release NB P-state force.
9. Program DRAM phy for power savings. See 2.9.7.10.
10. Program DCT for normal operation. See 2.9.7.7.

The DRAM subsystem is ready for use.

#### 2.9.7.1 Low Voltage DDR3

The processor supports JEDEC defined DDR3L and DDR3U devices which operate at voltages lower than 1.5V.

Platforms supporting low voltage devices should power up VDDIO at 1.35V. BIOS should not operate DIMMs at voltages higher than supported as indicated by SPD Byte 6: Module Nominal Voltage, VDD.

BIOS should consult vendor data sheets for the supply voltage regulator programming requirements. On supported platforms, BIOS must take steps to configure the supply voltage regulators as follows:

1. Read the SPD of all DIMMs within the programmable VDDIO domain and check all of the defined bits

- within the SPD byte to determine the common operating voltages.
- Configure VDDIO to match the lowest common supported voltage based on the SPD values.
    - If the DIMMs do not specify a common operating voltage then BIOS must take platform vendor defined action to notify the end user of the mismatch and to protect DIMMs from damage.
  - Additional derating of the DDR speed may be necessary for reliable operation at lower voltage. See [2.9.2 \[DCT Frequency Support\]](#).

### 2.9.7.2 NB P-state Specific Configuration

A subset of DCT configuration and training must be repeated for each enabled NB P-state. To accomplish this, BIOS forces the processor to the desired NB P-state and releases the force once DRAM initialization and training is complete. See [2.5.4.1 \[NB P-states\]](#) and [2.5.4.1.2.2 \[NB P-state Transitions\]](#).

### 2.9.7.3 Memory P-state Specific Configuration

A subset of DCT configuration and training must be repeated for each enabled memory P-state. See also [2.5.7.1 \[Memory P-states\]](#).

To accomplish this, BIOS forces the processor to the NBP0 state and uses the M0 context to train for values that are used by all memory P-states. For example, BIOS uses [D18F2x94\\_dct\[0\]\[MemClkFreq\]](#) to specify the dram frequency during each pass of training. BIOS saves the trained configuration values at each step and uses them appropriately as outlined below:

- Force NB P-state to NBP0. See [2.9.7.2](#).
- Program [D18F1x10C\[MemPsSel\]=0](#). See [2.9.3](#).
- Program [D18F2x9C\\_x0D04\\_E008\\_dct\[0\]\[PStateToAccess\]=0](#).
- Using the M0 context, BIOS optimally configures the controller and trains at DDR667 according to [2.9.7](#) steps 4 thru 7.
  - BIOS saves the trained values for use in M1 and for subsequent higher frequencies.
  - BIOS saves the controller configuration for use in M1.
  - BIOS optimally writes all the values to registers in step 6. It may optionally write the values as each is computed or trained. If this is done then BIOS must ensure that [D18F1x10C\[MemPsSel\]](#) and [D18F2x9C\\_x0D04\\_E008\\_dct\[0\]\[PStateToAccess\]](#) are configured correctly before and after each register access.
- BIOS increases the frequency and repeats step 4 until the target frequency is trained. The target frequency is always associated with M0.
- BIOS programs the M1 controller context with the DDR667 configuration and trained results. This must be done before the next subsequent NB P-state change.
  - Program [D18F1x10C\[MemPsSel\]=1](#).
  - Program [D18F2x9C\\_x0D04\\_E008\\_dct\[0\]\[PStateToAccess\] = 1](#).
  - Program [D18F2x2E0\\_dct\[0\]\[M1MemClkFreq\]](#) and [D18F2x9C\\_x0D0F\\_E000\\_dct\[0\]\\_mp\[1\]\[Rate, FreqValid\] = {04h, 1}](#).
  - Program DCT configuration and trained delay values saved earlier.
 

Three groups of registers must be written in the same order as the initialization sequence:

    - Fence values: [D18F2x9C\\_x0000\\_000C\\_dct\[0\]](#), [D18F2x9C\\_x0D0F\\_0\[F,8:0\]31\\_dct\[0\]](#), [D18F2x9C\\_x0D0F\\_E019\\_dct\[0\]](#).
    - Address/Command timing delays: [D18F2x9C\\_x0000\\_0004\\_dct\[0\]\\_mp\[1:0\]](#).
    - All other Data/Dqs delays: See step 6 in section [2.9.7](#).
  - Program [D18F2x9C\\_x0D04\\_E008\\_dct\[0\]\[PStateToAccess\] = 0](#).
  - Program [D18F1x10C\[MemPsSel\] = 0](#).

### 2.9.7.4 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 MEMCLK 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 `D18F2x9C_x0D0F_E013_dct[0] = 0118h`.
2. For each byte lane and each memory P-state: Program `D18F2x9C_x0D0F_0[F,8:0]13_dct[0]_mp[1:0][RxSsbMntClkEn] = 0`.
3. IF (`DdrRate >= 1600`) `D18F2x9C_x0D0F_0[F,8:0]38_dct[0][ReducedLoop] = 10b ENDIF`.
4. Program `D18F2xA8_dct[0][MemPhyPllPdMode] = 00b`.
5. Force the phy to M0 with the following sequence.
  - A. Program `D18F2x9C_x0D0F_E006_dct[0][PllLockTime] = 190h`.
  - B. Program `D18F2x9C_x0000_000B_dct[0] = 80800000h`.
  - C. Program `D18F2x9C_x0000_000D18F2x9C_x0000_000B_dct[0] = 40000000h`.
  - D. For each DCT: Program `D18F2x9C_x0000_000B_dct[0] = 80000000h`.
  - E. Program `D18F2x94_dct[0][MemClkFreq] = 200 MHz`.
  - F. Program `D18F2x94_dct[0][MemClkFreqVal] = 1`. Wait for `D18F2x94_dct[0][FreqChgInProg] = 0`.
6. Phy Voltage Level Programming. See 2.9.7.4.1.
7. DRAM channel frequency change. See 2.9.7.4.2.
8. Phy fence programming. See 2.9.7.4.3.
9. Phy compensation initialization. See 2.9.7.4.4.

#### 2.9.7.4.1 Phy Voltage Level Programming

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

- Program `D18F2x9C_x0D0F_0[F,8:0]1F_dct[0]_mp[1:0][RxVioLvl]`.
- Program `D18F2x9C_x0D0F_[C,8,2][2:0]1F_dct[0][RxVioLvl]`.
- Program `D18F2x9C_x0D0F_4009_dct[0][CmpVioLvl,ComparatorAdjust]`.
- Program `D18F2x9C_x0D0F_4006_dct[0][VrefSel] = 0`.
- Program `D18F2x9C_x0D0F_4007_dct[0]` per platform requirements.

See 2.9.7.1 [Low Voltage DDR3].

#### 2.9.7.4.2 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. BIOS performs the sequence only on DCTs with an enabled interface (`D18F2x94_dct[0][DisDramInterface]==0`):

For each DCT:

1. Program `D18F2x9C_x0D0F_E006_dct[0][PllLockTime] = 190h`.
2. Program `D18F2x94_dct[0][MemClkFreqVal] = 0`.
3. Program `D18F2x94_dct[0][MemClkFreq]` to the desired DRAM frequency.
4. Program the following parameters which must be configured prior to setting `MemClkFreqVal`:
  - `D18F2x94_dct[0][ProcOdtDis]`
  - `D18F2x9C_x0000_0004_dct[0]_mp[1:0]`
  - `D18F2x9C_x0D0F_0[F,8:0]13_dct[0]_mp[1:0][ProcOdtAdv]`
  - `D18F2x210_dct[0]_nbp[3:0][RdPtrInit, DataTxFifoWrDly]` of the current NB P-state for the target MEMCLK frequency. See also 2.9.7.2.
  - IF (target `DdrRate >= 1600`) `D18F2x9C_x0D0F_0[F,8:0]38_dct[0][ReducedLoop] = 10b ENDIF`.

For each DCT:

- Program `D18F2x94_dct[0][MemClkFreqVal]` = 1. Wait for `D18F2x94_dct[0][FreqChgInProg]` = 0.

For each DCT:

- Program `D18F2x9C_x0D0F_E006_dct[0][PllLockTime]` according to [Table 25](#).

BIOS must observe the following requirements:

- BIOS must not change the PLL frequency after DRAM has exited from self-refresh.
- BIOS must not change the PLL frequency after the DRAM training for DDR3 DIMMs is complete.

**Table 25: DDR PLL Lock Time**

<code>D18F2xA8_dct[0][MemPhyPllPdMode]</code>	<code>PllLockTime</code>
00b	0Fh
10b	190h

#### 2.9.7.4.2.1 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 [2.9.7.9.1 \[Write Levelization Training\]](#) and [2.9.7.9.2 \[DQS Receiver Enable Training\]](#).

- Force the NB P-state. See [2.9.7.2](#).
- Enter self-refresh:
  - Program `D18F2x90_dct[0][DisDllShutdownSR]` = 1.
  - Program `D18F2x90_dct[0][EnterSelfRef]` = 1.
  - Wait for `D18F2x90_dct[0][EnterSelfRef]` = 0.
- DRAM channel frequency change. See [2.9.7.4.2](#).
- Change NCLK frequency to meet NCLK-MEMCLK ratio requirements. See [2.5.3.1.5](#).
- Phy fence programming. See [2.9.7.4.3](#).
- Phy compensation initialization. See [2.9.7.4.4](#).
- Program SPD configuration. See [2.9.7.5](#).
- Program Non-SPD configuration. See [2.9.7.6](#).
- Exit self-refresh:
  - Program `D18F2x90_dct[0][ExitSelfRef]` = 1.
  - Wait for `D18F2x90_dct[0][ExitSelfRef]` = 0.
  - Program `D18F2x90_dct[0][DisDllShutdownSR]` = 0.
- Re-program devices with frequency dependent mode register field values. See [2.9.7.8](#).

#### 2.9.7.4.3 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 for each channel using the following steps:

- Program `D18F2x9C_x0D0F_0[F,8:0]31_dct[0]` = 0000\_0000h.
- Program `D18F2x9C_x0D0F_E019_dct[0]` = 0000\_0000h.
- Program `D18F2x9C_x0000_0008_dct[0]_mp[1:0][FenceTrSel]`=10b.
- Program `D18F2x9C_x0000_00[52:50]_dct[0]`=1313\_1313h.

5. Perform phy fence training. See 2.9.7.4.3.1 [Phy Fence Training].
6. Write the calculated fence value to `D18F2x9C_x0000_000C_dct[0][FenceThresholdTxDll]`.
7. Program `D18F2x9C_x0000_0008_dct[0]_mp[1:0][FenceTrSel]=01b`.
8. Program `D18F2x9C_x0000_00[52:50]_dct[0]=1313_1313h`.
9. Perform phy fence training. See 2.9.7.4.3.1 [Phy Fence Training].
10. Write the calculated fence value to `D18F2x9C_x0000_000C_dct[0][FenceThresholdRxDll]`.
11. Program `D18F2x9C_x0000_0008_dct[0]_mp[1:0][FenceTrSel]=11b`.
12. Program `D18F2x9C_x0000_00[52:50]_dct[0]=1313_1313h`.
13. Perform phy fence training. See 2.9.7.4.3.1 [Phy Fence Training].
14. Write the calculated fence value to `D18F2x9C_x0000_000C_dct[0][FenceThresholdTxPad]`.
15. Program Fence2 threshold for data as follows:
  - A. IF (`D18F2x9C_x0000_000C_dct[0][FenceThresholdTxPad] < 16`) THEN  
     `Fence2_TxPad[4:0] = { 1b, D18F2x9C_x0000_000C_dct[0][19:16]}`  
 ELSE  
     `Fence2_TxPad[4:0] = 00000b`  
 ENDIF.
  - B. IF (`D18F2x9C_x0000_000C_dct[0][FenceThresholdRxDll] < 16`) THEN  
     `Fence2_RxDll[4:0] = { 1b, D18F2x9C_x0000_000C_dct[0][24:21]}`  
 ELSE  
     `Fence2_RxDll[4:0] = 00000b`  
 ENDIF.
  - C. IF (`D18F2x9C_x0000_000C_dct[0][FenceThresholdTxDll] < 16`) THEN  
     `Fence2_TxDll[4:0] = { 1b, D18F2x9C_x0000_000C_dct[0][29:26]}`  
 ELSE  
     `Fence2_TxDll[4:0] = 00000b`  
 ENDIF.
  - D. Program `D18F2x9C_x0D0F_0[F,8:0]31_dct[0] = {000000b, Fence2_TxDll[4:0], Fence2_TxPad[4:0]}`.
  - E. Program `D18F2x9C_x0D0F_E019_dct[0] = {0b, Fence2_RxDll[4:0], 00000b, Fence2_TxPad[4:0]}`.
16. Reprogram `D18F2x9C_x0000_0004_dct[0]_mp[1:0]`.

When resuming from S3, BIOS reprograms `D18F2x9C_x0000_000C_dct[0][FenceThresholdTxDll]`, `FenceThresholdRxDll`, `FenceThresholdTxPad`, and `D18F2x9C_x0D0F_E019_dct[0]` from values stored in non-volatile storage instead of training. BIOS reprograms `D18F2x9C_x0D0F_0[F,8:0]31_dct[0]` and `D18F2x9C_x0D0F_E019_dct[0]` as indicated in the steps above.

BIOS may use `D18F2x9C_x0000_000C_dct[0]` and `D18F2x9C_x0D0F_E01A_dct[0]` for local storage of fence values during the intermediate time between training and writing the values into non-volatile storage in preparation for S3.

#### 2.9.7.4.3.1 Phy Fence Training

1. Program `D18F2x9C_x0000_0008_dct[0]_mp[1:0][PhyFenceTrEn]=1`.
2. Wait 2000 MEMCLKs.
3. Program `D18F2x9C_x0000_0008_dct[0]_mp[1:0][PhyFenceTrEn]=0`.
4. Read the phase recovery engine registers `D18F2x9C_x0000_00[52:50]_dct[0]`.
5. Calculate the average of the fine delay values of all byte lanes. If `FenceTrSel != 00b` subtract 8. If the result is negative then the fence value is zero.

#### 2.9.7.4.4 Phy Compensation Initialization

Each DDR IO driver has a programmable slew rate controlled by the pre-driver calibration code. The recom-



mended slew rate is a function of the DC drive strength. BIOS initializes the recommended nominal slew rate values as follows:

1. Program D18F2x9C\_x0000\_0008\_dct[0]\_mp[1:0][DisAutoComp] = 1.
2. Program D18F2x9C\_x0000\_0008\_dct[0]\_mp[1:0][DisablePredriverCal]=1.
3. Program TxPreP/TxPreN for Data and DQS according to Table 26, Table 27, or Table 28.
  - A. Program D18F2x9C\_x0D0F\_0[F,8:0]0[A,6]\_dct[0]={0000b, TxPreP, TxPreN}.
  - B. Program D18F2x9C\_x0D0F\_0[F,8:0]02\_dct[0]={1000b, TxPreP, TxPreN}.
4. Program TxPreP/TxPreN for Cmd/Addr according to Table 29, Table 30, or Table 31.
  - A. Program D18F2x9C\_x0D0F\_[C,8][1:0][12,0E,0A,06]\_dct[0]={0000b, TxPreP, TxPreN}.
  - B. Program D18F2x9C\_x0D0F\_[C,8][1:0]02\_dct[0]={1000b, TxPreP, TxPreN}.
5. Program TxPreP/TxPreN for Clock according to Table 32, Table 33, or Table 34.
  - A. Program D18F2x9C\_x0D0F\_2[2:0]02\_dct[0]={1000b, TxPreP, TxPreN}.
6. Program D18F2x9C\_x0000\_0008\_dct[0]\_mp[1:0][DisAutoComp] = 0.

**Table 26: Phy predriver calibration codes for Data/DQS at 1.5V**

DDR Rate	Drive Strength <sup>1</sup>	{TxPreP, TxPreN} <sup>2</sup>
667 - 2133	000b	FFFh
	001b	410h
	010b	208h
	011b	104h
1. IF (D18F2x9C_x0D0F_0[F,8:0]06) THEN See D18F2x9C_x0000_0000_dct[0]_mp[1:0][DqsDrvStren] ELSE See D18F2x9C_x0000_0000_dct[0]_mp[1:0][DataDrvStren] ENDIF. 2. See D18F2x9C_x0D0F_0[F,8:0]0[A,6]_dct[0] and D18F2x9C_x0D0F_0[F,8:0]02_dct[0].		

**Table 27: Phy predriver calibration codes for Data/DQS at 1.35V**

DDR Rate	Drive Strength <sup>1</sup>	{TxPreP, TxPreN} <sup>2</sup>
667 - 2133	000b	FFFh
	001b	FFFh
	010b	820h
	011b	410h
1. IF (D18F2x9C_x0D0F_0[F,8:0]06) THEN See D18F2x9C_x0000_0000_dct[0]_mp[1:0][DqsDrvStren] ELSE See D18F2x9C_x0000_0000_dct[0]_mp[1:0][DataDrvStren] ENDIF. 2. See D18F2x9C_x0D0F_0[F,8:0]0[A,6]_dct[0] and D18F2x9C_x0D0F_0[F,8:0]02_dct[0].		

**Table 28: Phy Predriver Calibration Codes for Data/DQS at 1.25V**

DDR Rate	Drive Strength <sup>1</sup>	{TxPreP, TxPreN} <sup>2</sup>
667 - 2133	000b	FFFh
	001b	FFFh
	010b	FFFh
	011b	FFFh

1. IF (D18F2x9C\_x0D0F\_0[F,8:0]06) THEN  
     See D18F2x9C\_x0000\_0000\_dct[0]\_mp[1:0][DqsDrvStren]  
 ELSE  
     See D18F2x9C\_x0000\_0000\_dct[0]\_mp[1:0][DataDrvStren]  
 ENDIF.

2. See D18F2x9C\_x0D0F\_0[F,8:0]0[A,6]\_dct[0] and D18F2x9C\_x0D0F\_0[F,8:0]02\_dct[0].

**Table 29: Phy predriver calibration codes for Cmd/Addr at 1.5V**

DDR Rate	Drive Strength <sup>1</sup>	{TxPreP, TxPreN} <sup>2</sup>
667 - 2133	101b	0C3h
	000b	082h
	001b	041h
	010b	041h
	011b	041h

1. IF (D18F2x9C\_x0D0F\_C002) THEN  
     See D18F2x9C\_x0000\_0000\_dct[0]\_mp[1:0][CkeDrvStren]  
 ELSEIF (D18F2x9C\_x0D0F\_800[A,6,2]) THEN  
     See D18F2x9C\_x0000\_0000\_dct[0]\_mp[1:0][CsOdtDrvStren]  
 ELSE  
     See D18F2x9C\_x0000\_0000\_dct[0]\_mp[1:0][AddrCmdDrvStren]  
 ENDIF.

2. See D18F2x9C\_x0D0F\_[C,8][1:0][12,0E,0A,06]\_dct[0] and D18F2x9C\_x0D0F\_[C,8][1:0]02\_dct[0].

**Table 30: Phy predriver calibration codes for Cmd/Addr at 1.35V**

DDR Rate	Drive Strength <sup>1</sup>	{TxPreP, TxPreN} <sup>2</sup>
667 - 2133	101b	208h
	000b	186h
	001b	104h
	010b	0C3h
	011b	082h

1. IF (D18F2x9C\_x0D0F\_C002)THEN  
     See D18F2x9C\_x0000\_0000\_dct[0]\_mp[1:0][CkeDrvStren]  
 ELSEIF (D18F2x9C\_x0D0F\_800[A,6,2])THEN  
     See D18F2x9C\_x0000\_0000\_dct[0]\_mp[1:0][CsOdtDrvStren]  
 ELSE  
     See D18F2x9C\_x0000\_0000\_dct[0]\_mp[1:0][AddrCmdDrvStren]  
 ENDIF.

2. See D18F2x9C\_x0D0F\_[C,8][1:0][12,0E,0A,06]\_dct[0] and D18F2x9C\_x0D0F\_[C,8][1:0]02\_dct[0].

**Table 31: Phy Predriver Calibration Codes for Cmd/Addr at 1.25V**

DDR Rate	Drive Strength <sup>1</sup>	{TxPreP, TxPreN} <sup>2</sup>
667- 2133	101b	410h
	000b	30Ch
	001b	28Ah
	010b	208h
	011b	186h

1. IF (D18F2x9C\_x0D0F\_C002)THEN  
     See D18F2x9C\_x0000\_0000\_dct[0]\_mp[1:0][CkeDrvStren]  
 ELSEIF (D18F2x9C\_x0D0F\_800[A,6,2])THEN  
     See D18F2x9C\_x0000\_0000\_dct[0]\_mp[1:0][CsOdtDrvStren]  
 ELSE  
     See D18F2x9C\_x0000\_0000\_dct[0]\_mp[1:0][AddrCmdDrvStren]  
 ENDIF.

2. See D18F2x9C\_x0D0F\_[C,8][1:0][12,0E,0A,06]\_dct[0] and D18F2x9C\_x0D0F\_[C,8][1:0]02\_dct[0].

**Table 32: Phy predriver calibration codes for Clock at 1.5V**

DDR Rate	Drive Strength <sup>1</sup>	{TxPreP, TxPreN} <sup>2</sup>
667 - 2133	101b	FFFh
	000b	FFFh
	001b	FFFh
	010b	FFFh
	011b	820h

1. See D18F2x9C\_x0000\_0000\_dct[0]\_mp[1:0][ClkDrvStren].

2. See D18F2x9C\_x0D0F\_2[2:0]02\_dct[0].

**Table 33: Phy predriver calibration codes for Clock at 1.35V**

DDR Rate	Drive Strength <sup>1</sup>	{TxPreP, TxPreN} <sup>2</sup>
667 - 2133	101b	FFFh
	000b	FFFh
	001b	FFFh
	010b	FFFh
	011b	FFFh
1. See D18F2x9C_x0000_0000_dct[0]_mp[1:0][ClkDrvStren]. 2. See D18F2x9C_x0D0F_2[2:0]02_dct[0].		

**Table 34: Phy Predriver Calibration Codes for Clock at 1.25V**

DDR Rate	Drive Strength <sup>1</sup>	{TxPreP, TxPreN} <sup>2</sup>
667 - 2133	101b	FFFh
	000b	FFFh
	001b	FFFh
	010b	FFFh
	011b	FFFh
1. See D18F2x9C_x0000_0000_dct[0]_mp[1:0][ClkDrvStren]. 2. See D18F2x9C_x0D0F_2[2:0]02_dct[0].		

### 2.9.7.5 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.

For solder-down DRAM, in the absence of an SPD ROM, BIOS provides the information necessary for DRAM configuration. For convenience, this document may refer to solder-down DRAM as a DIMM, and to DRAM data sheet or JEDEC DRAM specifications as SPD ROM based, unless otherwise noted.

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:

- D18F2x8C\_dct[0][Tref]
- D18F2x200\_dct[0]\_mp[1:0][Tras]: Active to precharge time
- D18F2x200\_dct[0]\_mp[1:0][Trp]: Precharge time
- D18F2x200\_dct[0]\_mp[1:0][Trcd]: RAS to CAS delay
- D18F2x200\_dct[0]\_mp[1:0][Tcl]: CAS latency
- D18F2x204\_dct[0]\_mp[1:0][Trtp]: Internal read to precharge command delay time

- D18F2x204\_dct[0]\_mp[1:0][FourActWindow]: Four activate window delay time
- D18F2x204\_dct[0]\_mp[1:0][Trrd]: Row active to row active delay
- D18F2x204\_dct[0]\_mp[1:0][Trc]: Active to active/refresh time
- D18F2x208\_dct[0][Trfc3, Trfc2, Trfc1, Trfc0]: Refresh recovery delay time
- D18F2x20C\_dct[0]\_mp[1:0][Twtr]: Internal write to read command delay time
- D18F2x22C\_dct[0]\_mp[1:0][Twr]: Write recovery time

Optimal cycle time is specified for each DIMM and is used to limit or determine bus frequency. See [2.9.7.8 \[DRAM Device and Controller Initialization\]](#).

The DRAM data sheet or the JEDEC DRAM specification provides values for some DRAM timing parameters that are required by the DCT, regardless of whether the DRAMs are solder-down. Most of these parameters have a fixed time component as part of their specification. If memory P-states are supported (See [2.5.7.1 \[Memory P-states\]](#)) and a timing register does not have per memory P-state contexts, BIOS must evaluate the minimum value for each frequency and choose the most pessimistic value for all frequencies to program into the register. These parameters are:

- D18F2x20C\_dct[0]\_mp[1:0][Tcwl]: CAS write latency
- D18F2x220\_dct[0][Tmrd]: MRS command cycle time
- D18F2x220\_dct[0][Tmod]: MRS command recovery time
- D18F2x224\_dct[0][Tzqcs]: Short calibration command recovery time
- D18F2x224\_dct[0][Tzqoper]: Long calibration command recovery time
- D18F2x248\_dct[0]\_mp[1:0][Txpdll]: Exit precharge power down (with DLL frozen) to command delay.
- D18F2x248\_dct[0]\_mp[1:0][Txp]: Exit precharge power down to command delay.
- D18F2x24C\_dct[0][Tcksrx]: Clock stable to self refresh exit delay
- D18F2x24C\_dct[0][Tcksre]: Self refresh entry to clock removal delay
- D18F2x24C\_dct[0][Tckesr]: Self refresh entry to command delay
- D18F2x24C\_dct[0][Tpd]: Power down entry to exit delay. BIOS may increase this value above the JEDEC minimum for certain power down modes.

### 2.9.7.6 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 [2.9.7.9 \[DRAM Training\]](#).
- Read and write latency differences.
- The phy's idle clock requirements on the data bus.
- DDR3 ODT timing requirements.
- NCLK frequency
- MEMCLK frequency

The following sub-sections 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) = D18F2x200\_dct[0]\_mp[1:0][Tcl] - D18F2x20C\_dct[0]\_mp[1:0][Tcwl].
- Read ODT Delay (ROD) = MAX(0, D18F2x240\_dct[0]\_mp[1:0][RdOdtOnDuration] - 6).
- Write ODT Delay (WOD) = MAX(0, D18F2x240\_dct[0]\_mp[1:0][WrOdtOnDuration] - 6).

- $WrEarly = ABS(D18F2x20C\_dct[0\_mp[1:0]][WrDqDqsEarly]) / 2$

### 2.9.7.6.1 TrdrdSdSc, TrdrdSdDc, and TrdrdDd (Read to Read Timing)

The optimal values for  $D18F2x218\_dct[0\_mp[1:0]][TrdrdSdSc, TrdrdSdDc, TrdrdDd]$  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:

- $TrdrdSdSc$  (in MEMCLKs) = 1.
- $TrdrdSdDc$  (in MEMCLKs) =  $MAX(TrdrdSdSc, 3 + (IF (D18F2xA8\_dct[0][PerRankTimingEn]) THEN CEIL(CDD_{TrdrdSdDc} / 2 + 0.5) ELSE 0 ENDIF))$ .
- $TrdrdDd$  (in MEMCLKs) =  $MAX(TrdrdSdDc, ROD + 3, CEIL(CDD_{TrdrdDd} / 2 + 3.5))$ .

The Critical Delay Difference (CDD) is the largest delay difference of the channel.

- Each delay difference is  $D18F2x9C\_x0000\_00[2A:10]\_dct[0\_mp[1:0]][DqsRcvEnGrossDelay]$  minus  $D18F2x9C\_x0000\_00[2A:10]\_dct[0\_mp[1:0]][DqsRcvEnGrossDelay]$ .
- For  $CDD_{TrdrdSdDc}$ , the subtraction terms are the delays of different chip selects within the same DIMM within the same byte lane.
- For  $CDD_{TrdrdDd}$ , the subtraction terms are the delays of different DIMMs within the same byte lane.

### 2.9.7.6.2 TwrwrSdSc, TwrwrSdDc, TwrwrDd (Write to Write Timing)

The optimal values for  $D18F2x214\_dct[0\_mp[1:0]][TwrwrSdSc, TwrwrSdDc, TwrwrDd]$  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:

- $TwrwrSdSc$  (in MEMCLKs) = 1.
- $TwrwrSdDc$  (in MEMCLKs) =  $MAX(TwrwrSdSc, WOD + 3, 3 + (IF (D18F2xA8\_dct[0][PerRankTimingEn]) THEN CEIL(CDD_{TwrwrSdDc} / 2 + 0.5) ELSE 0 ENDIF))$ .
- $TwrwrDd$  (in MEMCLKs) =  $MAX(TwrwrSdDc, WOD + 3, CEIL(CDD_{TwrwrDd} / 2 + 3.5))$ .

The Critical Delay Difference (CDD) is the largest delay difference of the channel.

- Each delay difference is  $D18F2x9C\_x0000\_00[4A:30]\_dct[0\_mp[1:0]][WrDqsGrossDly]$  minus  $D18F2x9C\_x0000\_00[4A:30]\_dct[0\_mp[1:0]][WrDqsGrossDly]$ .
- For  $CDD_{TwrwrSdDc}$ , the subtraction terms are the delays of different chip selects within the same DIMM within the same byte lane.
- For  $CDD_{TwrwrDd}$ , the subtraction terms are the delays of different DIMMs within the same byte lane.

### 2.9.7.6.3 Twrrd (Write to Read DIMM Termination Turn-around)

The optimal value for  $D18F2x218\_dct[0\_mp[1:0]][Twrrd]$  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:

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

The Critical Delay Difference (CDD) is the largest delay difference of the channel.

- Each delay difference is  $D18F2x9C\_x0000\_00[4A:30]\_dct[0\_mp[1:0]][WrDqsGrossDly]$  minus  $D18F2x9C\_x0000\_00[2A:10]\_dct[0\_mp[1:0]][DqsRcvEnGrossDelay]$ .

- For  $CDD_{Trwrtd}$ , the subtraction terms are the delays of different chip selects within the same byte lane.

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

The optimal value for  $D18F2x21C\_dct[0]_{mp}[1:0][TrwtTO]$  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) =  $MAX(ROD, CEIL(CDD_{TrwtTO} / 2 + WrEarly)) + LD + 3$ .
  - If 1 DIMM/ch, substitute  $ROD = 0$  in the above equation.

The Critical Delay Difference (CDD) is the largest delay difference of the channel.

- Each delay difference is  $D18F2x9C\_x0000\_00[2A:10]_dct[0]_{mp}[1:0][DqsRcvEnGrossDelay]$  minus  $D18F2x9C\_x0000\_00[4A:30]_dct[0]_{mp}[1:0][WrDqsGrossDly]$ .
- For  $CDD_{TrwtTO}$ , the subtraction terms are the delays of all chip selects within the same byte lane.

#### 2.9.7.6.5 DRAM ODT Control

BIOS configures the DIMM ODT behavior per chip select according to the DIMM population. The ODT pin patterns for reads and writes are programmed using  $D18F2x[234:230]_dct[0]$  and  $D18F2x[23C:238]_dct[0]$ , respectively.

BIOS also configures the ODT pin pattern used during write leveling by programming  $D18F2x9C\_x0000\_0008\_dct[0]_{mp}[1:0][WrLvOdtEn, WrLvOdt]$ . BIOS programs  $WrLvOdt$  with the  $D18F2x[23C:238]_dct[0]$  value provided for writes to the rank targeted by training. See 2.9.7.9.1 [Write Levelization Training].

Table 35: DDR3 ODT Pattern NumDimmSlots=1

DIMM0 <sup>1</sup>	D18F2x[234:230]_dct[0]		D18F2x[23C:238]_dct[0]	
	D18F2x234	D18F2x230	D18F2x23C	D18F2x238
SR	0000_0000h	0000_0000h	0000_0000h	0000_0001h
DR	0000_0000h	0000_0000h	0000_0000h	0000_0201h

1. DIMM0: MEMCSx[1:0], MEMODTx[1,0].

Table 36: DDR3 ODT Pattern NumDimmSlots=2

DIMM0 <sup>1</sup>	DIMM1 <sup>1</sup>	D18F2x[234:230]_dct[0]		D18F2x[23C:238]_dct[0]	
		D18F2x234	D18F2x230	D18F2x23C	D18F2x238
-	SR	0000_0000h	0000_0000h	0000_0000h	0004_0000h
-	DR	0000_0000h	0000_0000h	0000_0000h	0804_0000h
SR/DR	SR/DR	0000_0000h	0101_0404h	0000_0000h	0905_0605h

1. DIMM0: MEMCSx[1:0], MEMODTx[1,0].  
DIMM1: MEMCSx[3:2], MEMODTx[3,2].

**2.9.7.6.5.1 DRAM ODT Configuration**

These tables document the optimal settings for motherboards which meet the relevant AMD motherboard design guidelines. See 2.9.2 [DCT Frequency Support] for an overview of the DIMM population and memory bus speed support.

Table 37: BIOS recommendations for MR1[RttNom] and MR2[RttWr] **UDIMM FT3 Package**

Condition					MR1	MR2
NumDimmSlots	DdrRate	VDDIO	DIMM0	DIMM1	RttNom	RttWr
1	667, 800, 1066	1.5	SR, DR	-	010b	00b
1	1333, 1600, 1866	1.5	SR, DR	-	001b	00b
2	667, 800, 1066	1.5	NP	SR, DR	010b	00b
2	667, 800	1.5	SR, DR	SR, DR	011b	10b
2	1066, 1333	1.5	SR, DR	SR, DR	101b	10b
2	1333, 1600, 1866	1.5	NP	SR, DR	001b	00b
2	1600	1.5	SR, DR	SR, DR	100b	01b

Table 38: BIOS recommendations for MR1[RttNom] and MR2[RttWr] **SODIMM FT3, FS1b Package**

Condition					MR1	MR2
NumDimmSlots	DdrRate	VDDIO	DIMM0	DIMM1	RttNom	RttWr
1	667, 800, 1066	1.5, 1.35, 1.25	SR, DR	-	010b	00b
1	1333	1.5, 1.35, 1.25	SR, DR	-	001b	00b
1	1600	1.5, 1.35	SR, DR	-	001b	00b
2	667, 800, 1066	1.5, 1.35, 1.25	NP	SR, DR	010b	00b
2	667, 800	1.5, 1.35, 1.25	SR, DR	SR, DR	011b	10b
2	1066, 1333	1.5, 1.35, 1.25	SR, DR	SR, DR	101b	10b
2	1333	1.5, 1.35, 1.25	NP	SR, DR	001b	00b
2	1600	1.5, 1.35	NP	SR, DR	001b	00b
2	1600	1.5	SR, DR	SR, DR	100b	01b
2	1600	1.35	SR	SR	100b	01b



Table 39: BIOS recommendations for MR1[RttNom] and MR2[RttWr] SODIMM plus Solder-down DRAM FT3 Package

Condition					MR1	MR2
NumDimmSlots	DdrRate	VDDIO	DIMM0	DIMM1	RttNom	RttWr
2	667, 800, 1066	1.5, 1.35, 1.25	NP	SR, DR	010b	00b
2	667, 800	1.5, 1.35, 1.25	SR, DR	SR, DR	011b	10b
2	1066	1.5, 1.35, 1.25	SR, DR	SR, DR	101b	10b
2	1333	1.5, 1.35	NP	SR, DR	001b	00b
2	1333	1.5	SR, DR	SR, DR	101b	10b
2	1333	1.35	SR	SR	101b	10b

Table 40: BIOS recommendations for MR1[RttNom] and MR2[RttWr] Solder-down DRAM FT3 Package

Condition				MR1	MR2
NumDimmSlots	DdrRate	VDDIO	DIMM0	RttNom	RttWr
1	667	1.5, 1.35, 1.25	SR, DR	000b	00b
1	800, 1066	1.5, 1.35, 1.25	SR, DR	010b	00b
1	1333	1.5, 1.35, 1.25	SR, DR	001b	00b
1	1600	1.5, 1.35	SR, DR	001b	00b
1	1866	1.5	SR, DR	001b	00b

Table 41: BIOS recommendations for MR1[RttNom] and MR2[RttWr] UDIMM FS1b Package

Condition					MR1	MR2
NumDimmSlots	DdrRate	VDDIO	DIMM0	DIMM1	RttNom	RttWr
1	667, 800, 1066	1.5, 1.35, 1.25	SR, DR	-	010b	00b
1	1333, 1600	1.5, 1.35, 1.25	SR, DR	-	001b	00b
2	667, 800, 1066	1.5, 1.35, 1.25	NP	SR, DR	010b	00b
2	667, 800	1.5, 1.35, 1.25	SR, DR	SR, DR	011b	10b

Table 41: BIOS recommendations for MR1[RttNom] and MR2[RttWr] **UDIMM** FS1b Package

Condition					MR1	MR2	
NumDimmSlots	DdrRate	VDDIO	DIMM0	DIMM1	RttNom	RttWr	
	2	1066, 1333	1.5, 1.35, 1.25	SR, DR	SR, DR	101b	10b
	2	1333, 1600	1.5, 1.35, 1.25	NP	SR, DR	001b	00b
	2	1600	1.5, 1.35	SR, DR	SR, DR	100b	01b

#### 2.9.7.6.6 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. The following tables document the address timing and output driver settings on a per channel basis for DDR3. These tables document the optimal settings for motherboards which meet the relevant motherboard design guidelines. See 2.9.2 [DCT Frequency Support] for an overview of the DIMM and memory bus speed support.

When POdtOff=0, in all cases the processor ODT is off for writes and is on for reads.

Table 42: BIOS Recommendations for SlowAccessMode, POdtOff, Addr/Cmd Timings and Output Driver Control **UDIMM** FT3 Package

Condition					D18F2x9C_x0D0F_0[F:8:0]04_dct[0]_mp[1:0]		D18F2x9C_x0000_0004_dct[0]_mp[1:0]		D18F2x9C_x0000_0000_dct[0]_mp[1:0]	
NumDimmSlots	DdrRate	VDDIO	DIMM0	DIMM1	D18F2x94_dct[0]		POdtOff	SlowAccessMode		
1	667, 800	1.5	SR, DR	-	0	0	00000000h	00002222h		
1	1066	1.5	SR	-	0	0	003D3D3Dh	10002222h		
1	1066	1.5	DR	-	0	0	00000000h	10002222h		
1	1333	1.5	SR	-	0	0	003D3D3Dh	20112222h		
1	1333	1.5	DR	-	0	0	00003D3Dh	20112222h		
1	1600, 1866	1.5	SR	-	0	0	003C3C3Ch	30332222h		
1	1600, 1866	1.5	DR	-	1	0	00003C3Ch	30332222h		
2	667, 800	1.5	NP	SR, DR	0	0	00000000h	00002222h		
2	667	1.5	SR, DR	SR, DR	1	0	00000000h	10222323h		
2	800	1.5	SR, DR	SR, DR	1	0	00000000h	20222323h		
2	1066	1.5	NP	SR	0	0	003D3D3Dh	10002222h		
2	1066	1.5	NP	DR	0	0	00000000h	10002222h		
2	1066, 1333, 1600	1.5	SR, DR	SR, DR	1	0	00000000h	30222323h		
2	1333	1.5	NP	SR	0	0	003D3D3Dh	20112222h		
2	1333	1.5	NP	DR	0	0	00003D3Dh	20112222h		

Table 42: BIOS Recommendations for SlowAccessMode, POdtOff, Addr/Cmd Timings and Output Driver Control **UDIMM** FT3 Package

Condition					D18F2x9C_x0D0F_0[F;8:0]04_dct[0]_mp[1:0]	D18F2x9C_x0000_0004_dct[0]_mp[1:0]	D18F2x9C_x0000_0000_dct[0]_mp[1:0]
NumDimmSlots	DdrRate	VDDIO	DIMM0	DIMM1	SlowAccessMode	POdtOff	
2	1600, 1866	1.5	NP	SR	0	0	003C3C3Ch 30332222h
2	1600, 1866	1.5	NP	DR	1	0	00003C3Ch 30332222h

Table 43: BIOS Recommendations for SlowAccessMode, POdtOff, Addr/Cmd Timings and Output Driver Control SODIMM FT3, FS1b Package

Condition					D18F2x9C_x0D0F_0[F:8:0]04_dct[0]_mp[1:0]		D18F2x94_dct[0]		D18F2x9C_x0000_0004_dct[0]_mp[1:0]		D18F2x9C_x0000_0000_dct[0]_mp[1:0]	
NumDimmSlots	DdrRate	VDDIO	DIMM0	DIMM1	SlowAccessMode	POdtOff						
1	667, 800	1.5, 1.35, 1.25	SR, DR	-	0	0	00000000h		00002222h			
1	1066	1.5, 1.35, 1.25	SR	-	0	0	003D3D3Dh		10002222h			
1	1066	1.5, 1.35, 1.25	DR	-	0	0	00000000h		10002222h			
1	1333	1.5, 1.35, 1.25	SR	-	0	0	003D3D3Dh		20112222h			
1	1333	1.5, 1.35, 1.25	DR	-	0	0	00003D3Dh		20112222h			
1	1600	1.5, 1.35	SR	-	0	0	003C3C3Ch		30332222h			
1	1600	1.5, 1.35	DR	-	1	0	00003C3Ch		30332222h			
2	667, 800	1.5, 1.35, 1.25	NP	SR, DR	0	0	00000000h		00002222h			
2	667	1.5, 1.35, 1.25	SR, DR	SR, DR	1	0	00000000h		10222323h			
2	800	1.5, 1.35, 1.25	SR, DR	SR, DR	1	0	00000000h		20222323h			
2	1066	1.5, 1.35, 1.25	NP	SR	0	0	003D3D3Dh		10002222h			
2	1066	1.5, 1.35, 1.25	NP	DR	0	0	00000000h		10002222h			
2	1066, 1333	1.5, 1.35, 1.25	SR, DR	SR, DR	1	0	00000000h		30222323h			
2	1333	1.5, 1.35, 1.25	NP	SR	0	0	003D3D3Dh		20112222h			
2	1333	1.5, 1.35, 1.25	NP	DR	0	0	00003D3Dh		20112222h			
2	1600	1.5, 1.35	NP	SR	0	0	003C3C3Ch		30332222h			
2	1600	1.5, 1.35	NP	DR	1	0	00003C3Ch		30332222h			
2	1600	1.5	SR, DR	SR, DR	1	0	00000000h		30222323h			

Table 43: BIOS Recommendations for SlowAccessMode, POdtOff, Addr/Cmd Timings and Output Driver Control **SODIMM** FT3, FS1b Package

Condition					D18F2x9C_x0D0F_0[F;8:0]04_dct[0]_mpl[1:0]	D18F2x9C_x0000_0004_dct[0]_mpl[1:0]	D18F2x9C_x0000_0000_dct[0]_mpl[1:0]
NumDimmSlots	DdrRate	VDDIO	DIMM0	DIMM1	SlowAccessMode	POdtOff	
2	1600	1.35	SR	SR	1	0	00000000h 30222323h

Table 44: BIOS Recommendations for SlowAccessMode, POdtOff, Addr/Cmd Timings and Output Driver Control SODIMM plus Solder-down DRAM FT3 Package

Condition					D18F2x94_dct[0]	D18F2x9C_x0D0F_0[F:8:0]04_dct[0]_mp[1:0]	D18F2x9C_x0000_0004_dct[0]_mp[1:0]	D18F2x9C_x0000_0000_dct[0]_mp[1:0]
NumDimmSlots	DdrRate	VDDIO	DIMM0	DIMM1	SlowAccessMode	POdtOff		
2	667, 800	1.5, 1.35, 1.25	NP	SR, DR	0	0	00000000h	00002222h
2	667	1.5, 1.35, 1.25	SR, DR	SR, DR	1	0	00000000h	10222323h
2	800	1.5, 1.35, 1.25	SR, DR	SR, DR	1	0	00000000h	20222323h
2	1066	1.5, 1.35, 1.25	NP	SR	0	0	003D3D3Dh	10002222h
2	1066	1.5, 1.35, 1.25	NP	DR	0	0	00000000h	10002222h
2	1066	1.5, 1.35, 1.25	SR, DR	SR, DR	1	0	00000000h	30222323h
2	1333	1.5, 1.35	NP	SR	0	0	003D3D3Dh	20112222h
2	1333	1.5, 1.35	NP	DR	0	0	00003D3Dh	20112222h
2	1333	1.5	SR, DR	SR, DR	1	0	00000000h	30222323h
2	1333	1.35	SR	SR	1	0	00000000h	30222323h

Table 45: BIOS Recommendations for SlowAccessMode, POdtOff, Addr/Cmd Timings and Output Driver Control Solder-down DRAM FT3 Package

Condition				D18F2x9C_x0DD0F_0[F:8:0]04_dct[0]_mp[1:0] D18F2x94_dct[0]		D18F2x9C_x0000_0004_dct[0]_mp[1:0]		D18F2x9C_x0000_0000_dct[0]_mp[1:0]	
NumDimmSlots	DdrRate	VDDIO	DIMM0	SlowAccessMode	POdtOff				
1	667	1.5, 1.35, 1.25	SR, DR	0	1	00000000h	00000000h		
1	800	1.5, 1.35, 1.25	SR, DR	0	0	00000000h	00000000h		
1	1066	1.5, 1.35, 1.25	SR	0	0	003D3D3Dh	10000000h		
1	1066	1.5, 1.35, 1.25	DR	0	0	00000000h	10000000h		
1	1333	1.5, 1.35, 1.25	SR	0	0	003D3D3Dh	20000000h		
1	1333	1.5, 1.35, 1.25	DR	0	0	00003D3Dh	20110000h		
1	1600	1.5, 1.35	SR	0	0	003C3C3Ch	30110000h		
1	1600	1.5, 1.35	DR	1	0	00003C3Ch	30110000h		



Table 46: BIOS Recommendations for SlowAccessMode, POdtOff, Addr/Cmd Timings and Output Driver Control **UDIMM** FS1b Package

Condition					D18F2x9C_x0D0F_0[F:8:0]04_dct[0]_mp[1:0]		D18F2x94_dct[0]		D18F2x9C_x0000_0004_dct[0]_mp[1:0]		D18F2x9C_x0000_0000_dct[0]_mp[1:0]	
NumDimmSlots	DdrRate	VDDIO	DIMM0	DIMM1	SlowAccessMode	POdtOff						
1	667, 800	1.5, 1.35, 1.25	SR, DR	-	0	0	00000000h		00002222h			
1	1066	1.5, 1.35, 1.25	SR	-	0	0	003D3D3Dh		10002222h			
1	1066	1.5, 1.35, 1.25	DR	-	0	0	00000000h		10002222h			
1	1333	1.5, 1.35, 1.25	SR	-	0	0	003D3D3Dh		20112222h			
1	1333	1.5, 1.35, 1.25	DR	-	0	0	00003D3Dh		20112222h			
1	1600	1.5, 1.35, 1.25	SR	-	0	0	003C3C3Ch		30332222h			
1	1600	1.5, 1.35, 1.25	DR	-	1	0	00003C3Ch		30332222h			
2	667, 800	1.5, 1.35, 1.25	NP	SR, DR	0	0	00000000h		00002222h			
2	667	1.5, 1.35, 1.25	SR, DR	SR, DR	1	0	00000000h		10222323h			
2	800	1.5, 1.35, 1.25	SR, DR	SR, DR	1	0	00000000h		20222323h			
2	1066	1.5, 1.35, 1.25	NP	SR	0	0	003D3D3Dh		10002222h			
2	1066	1.5, 1.35, 1.25	NP	DR	0	0	00000000h		10002222h			
2	1066, 1333	1.5, 1.35, 1.25	SR, DR	SR, DR	1	0	00000000h		30222323h			
2	1333	1.5, 1.35, 1.25	NP	SR	0	0	003D3D3Dh		20112222h			
2	1333	1.5, 1.35, 1.25	NP	DR	0	0	00003D3Dh		20112222h			
2	1600	1.5, 1.35, 1.25	NP	SR	0	0	003C3C3Ch		30332222h			
2	1600	1.5, 1.35, 1.25	NP	DR	1	0	00003C3Ch		30332222h			
2	1600	1.5, 1.35	SR, DR	SR, DR	1	0	00000000h		30222323h			

### 2.9.7.7 DCT Training Specific Configuration

The DCT requires certain features be disabled during DRAM device initialization and training. BIOS should program the registers in [Table 47](#) before DRAM device initialization and training. For normal operation, BIOS programs the recommended values if provided in [Table 47](#). BIOS must quiesce all other forms of DRAM traffic on the channel being trained. See [2.9.7 \[DCT/DRAM Initialization and Resume\]](#).

**Table 47: DCT Training Specific Register Values**

Register	Training	Normal Operation
D18F2x78_dct[0][AddrCmdTriEn]	0	1
D18F2x8C_dct[0][DisAutoRefresh]	1	0
D18F2x90_dct[0][ForceAutoPchg]	0	0
D18F2x90_dct[0][DynPageCloseEn]	0	0
D18F2x94_dct[0][BankSwizzleMode]	0	1
D18F2x94_dct[0][DcqBypassMax]	0	1Fh
D18F2x94_dct[0][PowerDownEn]	0	1
D18F2x94_dct[0][ZqcsInterval]	00b	10b
D18F2x9C_x0000_000D_dct[0]_mp[1:0][RxMax-DurDllNoLock]	000b	See <a href="#">2.9.7.10</a>
D18F2x9C_x0000_000D_dct[0]_mp[1:0][TxMax-DurDllNoLock]	000b	See <a href="#">2.9.7.10</a>
D18F2x9C_x0D0F_0[F,8:0]10_dct[0]_mp[1:0][EnRxPadStandby]	0	See <a href="#">2.9.7.10</a>
D18F2xA4[CmdThrottleMode]	000b	See <a href="#">2.9.12</a>
D18F2xA4[ODTSEn]	000b	See <a href="#">2.9.12</a>
D18F2xA4[BwCapEn]	0	See <a href="#">2.9.12</a>
D18F2xA8_dct[0][BankSwap]	0	1
D18F3x58[DramScrub] <sup>2</sup>	0	See <a href="#">2.14.1.8</a>
D18F3x5C[ScrubReDirEn] <sup>2</sup>	0	IF (D18F3x44[DramEc-cEn]==1) THEN 1 ELSE 0 ENDIF
<ol style="list-style-type: none"> <li>1. Programmed specific to the current platform or memory configuration.</li> <li>2. BIOS must quiesce all other forms of DRAM traffic on the channel when performing writes via D18F2x98_dct[0]. See also D18F3x88[DisDramScrub].</li> </ol>		

### 2.9.7.8 DRAM Device and Controller Initialization

BIOS initializes the DRAM devices and the controller using a software controlled sequence. See [2.9.7.8.1 \[Software DDR3 Device Initialization\]](#).

BIOS must observe additional requirements for changing the PLL frequency when setting D18F2x7C\_dct[0][EnDramInit]. See [2.9.7.4.2 \[DRAM Channel Frequency Change\]](#).

DRAM initialization is complete after the value of `D18F2x7C_dct[0][EnDramInit]` is written by BIOS from 1 to 0 in the software-controlled sequence.

### 2.9.7.8.1 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).

1. Program `D18F2x7C_dct[0][EnDramInit]` = 1.
2. Wait 200 us.
3. Program `D18F2x7C_dct[0][DeassertMemRstX]` = 1.
4. Wait 500 us.
5. Program `D18F2x7C_dct[0][AssertCke]` = 1.
6. Wait 360 ns.
7. Send MRS(2).
8. Send MRS(3). Ordinarily at this time, `MrsAddress[2:0]` = 000b.
9. Send MRS(1) with `MrsAddress[7]` = 0.
10. Send MRS(0) with `MrsAddress[8]` = 1.
11. Send two ZQCL commands.
- BIOS instructs the DCT to send a ZQCL command by programming `D18F2x7C_dct[0]` as follows:
  - Program `MrsAddress[10]` = 1.
  - Program `SendZQCmd` = 1.
  - Wait for `SendZQCmd` = 0.
  - Wait 512 MEMCLKs.
12. Program `D18F2x7C_dct[0][EnDramInit]` = 0.
13. Program `D18F2x2E8_dct[0]_mp[1:0]`, `D18F2x2EC_dct[0]_mp[1:0]` with a copy of the mode register data sent in the steps above, except with `MR0[DLL]` = 0 .

#### 2.9.7.8.1.1 DDR3 MR Initialization

BIOS instructs the DCT to send MRS commands by programming `D18F2x7C_dct[0]` as follows:

1. Program `MrsBank` and `MrsAddress` as specified below:
  - `MrsBank[2:0]` = BA2:BA0.
  - `MrsAddress[15:0]` = A15:A0.
  - BIOS may need to remap bits, see also `D18F2x[5C:40]_dct[0][OnDimmMirror]`.
  - Set all other bits in `MrsAddress` to zero. BIOS should write reserved fields as 0.
2. Program `MrsChipSel` as appropriate.
3. Program `SendMrsCmd` = 1.
4. Wait for `SendMrsCmd` = 0.

### MR0 DDR3 MR0

Table 48: BIOS Recommendations for `MR0[WR]`

Condition	MR0
<code>D18F2x22C_dct[0]_mp[1:0][Twr]</code>	<code>WR[3:0]</code>
10h	0000b
5h	0001b
6h	0010b

Table 48: BIOS Recommendations for MR0[WR]

Condition	MR0
D18F2x22C_dct[0]_mp[1:0][Twr]	WR[3:0]
7h	0011b
8h	0100b
Ah	0101b
Ch	0110b
Eh	0111b
12h	1001b

Table 49: BIOS Recommendations for MR0[CL[3:0]]

Condition	MR0
D18F2x200_dct[0]_mp[1:0][Tcl]	CL[3:0]
5h	2h
6h	4h
7h	6h
8h	8h
9h	Ah
Ah	Ch
Bh	Eh
Ch	1h
Dh	3h
Eh	5h
Fh	7h
10h	9h
11h	Bh

Bits	Description
15:14	Reserved.
13	<b>WR[3]: write recovery for autoprecharge.</b> See WR[2:0].
12	<b>PPD: DLL control for precharge powerdown.</b> BIOS: D18F2x84_dct[0][PchgPDMoSel].
11:9	<b>WR[2:0]: write recovery for autoprecharge.</b> WR[3:0] = {WR[3], WR[2:0]}. BIOS: Table 48.
8	<b>DLL: DLL reset.</b> BIOS: See 2.9.7.8.1.
7	<b>TM: test mode.</b> BIOS: 0.
6:4	<b>CL[3:1]. CAS latency.</b> See: CL[0].
3	<b>RBT: read burst type.</b> BIOS: 1.
2	<b>CL[0]. CAS latency.</b> CL[3:0] = {CL[3:1], CL[0]}. BIOS: Table 49.
1:0	<b>BL: burst length.</b> BIOS: D18F2x84_dct[0][BurstCtrl].

**MR1 DDR3 MR1**

Bits	Description
15:13	Reserved.
12	<b>Qoff: Qoff.</b> BIOS: See <a href="#">2.9.7.9.1</a> .
11	<b>TDQS: TDQS enable.</b> BIOS: 0.
10	Reserved.
9	<b>RttNom[2]: RttNom.</b> See: RttNom[0].
8	Reserved.
7	<b>Level: write leveling enable.</b> BIOS: See <a href="#">2.9.7.9.1</a> .
6	<b>RttNom[1]: RttNom.</b> See: RttNom[0].
5	<b>DIC[1]: output driver impedance control.</b> See: DIC[0].
4:3	<b>AL: additive latency.</b> BIOS: 0.
2	<b>RttNom[0]: RttNom.</b> RttNom[2:0] = {RttNom[2], RttNom[1], RttNom[0]}. BIOS: See <a href="#">2.9.7.6.5.1</a>
1	<b>DIC[0]: output driver impedance control.</b> DIC[1:0] = {DIC[1], DIC[0]}. BIOS: 01b.
0	<b>DLL: DLL enable.</b> BIOS: 0.

**MR2 DDR3 MR2**

Table 50: BIOS Recommendations for MR2[ASR, SRT]

Condition		MR2	
AutoSelfRefresh	ExtendedTemperature-Range	ASR	SRT
0	0	0	0
0	1	0	1
1	-	1	0

Bits	Description
15:11	Reserved.MR2
10:9	<b>RttWr: RttWr.</b> BIOS: See <a href="#">2.9.7.6.5.1</a> .
8	Reserved.
7	<b>SRT: self refresh temperature range.</b> BIOS: <a href="#">Table 50</a> .
6	<b>ASR: auto self refresh.</b> BIOS: <a href="#">Table 50</a> .
5:3	<b>CWL: CAS write latency.</b> BIOS: <a href="#">D18F2x20C_dct[0]_mp[1:0][Tcwl] - 5</a> .
2:0	<b>PASR: partial array self refresh.</b> BIOS: 0.

**MR3 DDR3 MR3**

Bits	Description
15:3	Reserved.
2	<b>MPR: MPR operation.</b> BIOS: 0.
1:0	<b>MPRLoc: MPR location.</b> BIOS: 0.

**2.9.7.9 DRAM Training**

This section describes detailed 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 [2.9.7.8 \[DRAM Device and Controller Initialization\]](#).

Some of the DRAM training steps described in this section require two passes if the target MEMCLK frequency is greater than 333 MHz. 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 [D18F2x94\\_dct\[0\]\[MemClkFreq\]](#).

Product specific training requirements are as follows:

- Program [D18F2x9C\\_x0000\\_0008\\_dct\[0\]\\_mp\[1:0\]\[TrNibbleSel\]=0](#).
- Program [D18F2xA8\\_dct\[0\]\[PerRankTimingEn\]=1](#).

See [2.9.7.7 \[DCT Training Specific Configuration\]](#) 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.

**2.9.7.9.1 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 rank basis. If the target frequency is greater than 333 MHz then BIOS performs two passes; otherwise, only one pass is required. See [2.9.7.4.2 \[DRAM Channel Frequency Change\]](#).

- Pass 1: Configure the memory subsystem for 333 MHz MEMCLK frequency.
- Pass 2: Configure the memory subsystem for the target MEMCLK frequency.

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

For each rank:

1. Prepare the DIMMs for write levelization using DDR3-defined MR commands.
  - A. Configure the output driver and on-die termination of the target DIMM as follows:
    - For the target 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 2.9.7.6.6 [DRAM Address Timing and Output Driver Compensation Control].
- B. Configure Rtt\_Nom on the non-target DIMMs as normal. See 2.9.7.6.6 [DRAM Address Timing and Output Driver Compensation Control].
2. Wait 40 MEMCLKs.
  3. Configure the phy for write levelization training:
    - A. Program D18F2x9C\_x0000\_0008\_dct[0]\_mp[1:0][WrtLvTrEn]=0.
    - B. Program D18F2x9C\_x0000\_0008\_dct[0]\_mp[1:0][TrChipSel] to specify the target rank to be trained.
    - C. Program D18F2x9C\_x0000\_0008\_dct[0]\_mp[1:0][WrLvOdt] to the proper ODT settings for the current memory subsystem configuration. See 2.9.7.6.5 [DRAM ODT Control].
    - D. Program D18F2x9C\_x0000\_0008\_dct[0]\_mp[1:0][WrLvOdtEn]=1.
    - E. MFENCE.
    - F. Wait 10 MEMCLKs to allow for ODT signal settling.
    - G. For each lane program an initial value to registers D18F2x9C\_x0000\_00[52:50]\_dct[0] to set the gross and fine delay. See 2.9.7.9.1.1 [Write Leveling Seed Value].
  4. Perform write leveling of the devices on the DIMM:
    - A. Program D18F2x9C\_x0000\_0008\_dct[0]\_mp[1:0][WrtLvTrEn]=1.
    - B. MFENCE.
    - C. Wait 200 MEMCLKs.
    - D. Program D18F2x9C\_x0000\_0008\_dct[0]\_mp[1:0][WrtLvTrEn]=0.
    - E. Read from registers D18F2x9C\_x0000\_00[52:50]\_dct[0] 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 D18F2x9C\_x0000\_0008\_dct[0]\_mp[1:0][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:
    - A. Configure all ranks of the target DIMM for normal operation.
    - B. Enable the output drivers of all ranks of the target DIMM.
    - C. For a two or more DIMM system, program the Rtt\_Nom value for the target DIMM to the normal operating termination.

For each rank:

- BIOS calculates and programs the final saved gross and fine delay values for each lane into D18F2x9C\_x0000\_00[4A:30]\_dct[0]\_mp[1:0] [DRAM DQS Write Timing].
  - $WrDqsFineDly = PhRecFineDly$ .
  - $GrossDly = SeedGross + PhRecGrossDly - SeedPreGross$ .
    - The Critical Gross Delay (CGD) is the minimum GrossDly of all byte lanes and all DIMMs.
    - If  $(CGD < 0)$  Then
      - $D18F2x20C\_dct[0]_mp[1:0][WrDqDqsEarly] = ABS(CGD)$
      - $WrDqsGrossDly = GrossDly + WrDqDqsEarly$
    - Else
      - $D18F2x20C\_dct[0]_mp[1:0][WrDqDqsEarly] = 1$ .
      - $WrDqsGrossDly = GrossDly + 1$ .

### 2.9.7.9.1.1 Write Leveling Seed Value

The seed value for pass 1 of write leveling is design and platform specific. The platform vendor may need to characterize and adjust this value for proper write levelization training. The seed delay value must fall within +/- 1/2 MEMCLK, including silicon PVT margin and jitter of the measured clock delay.

1. Calculate the total seed based on the following:
  - Pass 1: SeedTotal = configuration specific seed value found in [Table 51](#).
  - Pass N:
    - SeedTotalPreScaling = the total delay values obtained from the previous (N-1) pass of write levelization training.
      - Write Leveling Total Delay =  $D18F2x9C\_x0000\_00[4A:30]_{dct}[0]_{mp}[1:0][WrDqsGrossDly, WrDqsFineDly] - (0x20 * D18F2x20C\_dct[0]_{mp}[1:0][WrDqDqsEarly])$
      - SeedTotal = 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  $D18F2x9C\_x0000\_00[52:50]_{dct}[0][PhRecFineDly] = SeedFine$ .
6. Program  $D18F2x9C\_x0000\_00[52:50]_{dct}[0][PhRecGrossDly] = SeedPreGross$ .

**Table 51: DDR3 Write Leveling Seed Values**

DIMM Type	Seed Value <sup>1</sup>
Solder-down DRAM	0Eh
SODIMM	0Eh
UDIMM	15h
1. DDR3-667 (333 MHz).	

### 2.9.7.9.2 DQS Receiver Enable Training

Receiver enable delay training is used to dynamically determine the optimal delay value for [D18F2x9C\\_x0000\\_00\[2A:10\]\\_dct\[0\]\\_mp\[1:0\] \[DRAM DQS Receiver Enable Timing\]](#). 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.

Prior to DQS Receiver Enable Training, BIOS must program  $D18F2x210\_dct[0]_{nbp}[3:0][MaxRdLatency]$  based on the seeded value of  $D18F2x9C\_x0000\_00[2A:10]_{dct}[0]_{mp}[1:0]$ . See [2.9.7.9.5 \[Calculating MaxRdLatency\]](#).

Training is accomplished on a per channel, per rank basis. If the target frequency is greater than 333 MHz then BIOS performs two passes; otherwise only one pass is required. See [2.9.7.4.2 \[DRAM Channel Frequency Change\]](#).

- Pass 1: Configure the memory subsystem for 333 MHz MEMCLK frequency.
- Pass 2: Configure the memory subsystem for the target MEMCLK frequency.



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

For each rank:

1. Program `D18F2x9C_x0000_0008_dct[0]_mp[1:0][TrChipSel]` to specify the target rank to be trained.
2. For each lane program an initial value to registers `D18F2x9C_x0000_00[52:50]_dct[0]` to set the gross and fine delay as specified in 2.9.7.9.2.1 [DQS Receiver Enable Training Seed Value].
3. Program `D18F2x9C_x0000_0008_dct[0]_mp[1:0][DqsRcvTrEn]=1`.
4. Issue 192 read requests to the target rank. See 2.9.8 [Continuous Pattern Generation].
5. Program `D18F2x9C_x0000_0008_dct[0]_mp[1:0][DqsRcvTrEn]=0`.
6. Read `D18F2x9C_x0000_00[52:50]_dct[0][PhRecGrossDly, PhRecFineDly]` 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 `D18F2x9C_x0000_00[2A:10]_dct[0]_mp[1:0][DqsRcvEnGrossDelay, DqsRcvEnFineDelay]`. Save the result for use later.
  - $DqsRcvEnFineDelay = PhRecFineDly$ .
  - $DqsRcvEnGrossDelay = SeedGross + PhRecGrossDly - SeedPreGross + 1$ .

### 2.9.7.9.2.1 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 first expected rise edge of DQS 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.

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:  $SeedTotal = \text{The seed value found in Table 52} + \text{the total delay values obtained from the first pass of write levelization training. See 2.9.7.9.1 [Write Levelization Training].}$ 
    - $Write\ Leveling\ Total\ Delay = D18F2x9C_x0000_00[4A:30]_dct[0]_mp[1:0][WrDqsGrossDly, WrDqsFineDly] - (0x20 * D18F2x20C_dct[0]_mp[1:0][WrDqDqsEarly])$
  - Pass N:
    - $RegisterDelay = 0$
    - $SeedTotalPreScaling = (\text{the total delay values in } D18F2x9C_x0000_00[2A:10]_dct[0]_mp[1:0] \text{ from the previous (N-1) pass of training}) - RegisterDelay - 20h$ .
    - $SeedTotal = RegisterDelay + FLOOR(SeedTotalPreScaling * (\text{target frequency}) / (\text{frequency from previous pass}))$ .
2.  $SeedGross = SeedTotal \text{ DIV } 32$ .
3.  $SeedFine = SeedTotal \text{ MOD } 32$ .
4. If (SeedGross is odd)  
then  $SeedPreGross = 1$   
else  $SeedPreGross = 2$ .
5. Program `D18F2x9C_x0000_00[52:50]_dct[0][PhRecFineDly] = SeedFine`.
6. Program `D18F2x9C_x0000_00[52:50]_dct[0][PhRecGrossDly] = SeedPreGross`.
7. Program `D18F2x9C_x0000_00[2A:10]_dct[0]_mp[1:0][DqsRcvEnGrossDelay] = IF ((NBCOF / DdrRate < 1) && (D18F2x200_dct[0]_mp[1:0][Tcl]=5) && (SeedGross < 1) THEN 1 ELSE SeedGross ENDIF`.

**Table 52. DDR3 DQS Receiver Enable Training Seed Values**

DIMM Type	Seed Value <sup>1</sup>
Solder-down DRAM	20h
SODIMM	32h
UDIMM	32h
1. DDR3-667 (333 MHz)	

### 2.9.7.9.3 DQS Receiver Enable Cycle Training

Receiver enable delay cycle training is used to train the gross delay settings of `D18F2x9C_x0000_00[2A:10]_dct[0]_mp[1:0]` to the middle of the received read preamble using the phy phase results.

For each rank and lane:

1. Program `D18F2x9C_x0D0F_0[F,8:0]30_dct[0][BlockRxDqsLock] = 1`.
2.  $RxEnOrig = D18F2x9C\_x0000\_00[2A:10]_dct[0]_mp[1:0][DqsRcvEnGrossDelay, DqsRcvEnFineDelay]$  result from 2.9.7.9.2 [DQS Receiver Enable Training].
3.  $RxEnOffset = MOD(RxEnOrig + 10h, 40h)$
4. For each `DqsRcvEn` value beginning from `RxEnOffset` incrementing by 1 MEMCLK:
  - A. Program `D18F2x9C_x0000_00[2A:10]_dct[0]_mp[1:0][DqsRcvEnGrossDelay, DqsRcvEnFineDelay]` with the current value.
  - B. Perform 2.9.7.9.4 [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 `D18F2x9C_x0000_00[2A:10]_dct[0]_mp[1:0][DqsRcvEnGrossDelay, DqsRcvEnFineDelay] = DqsRcvEnCycle - 10h`.
6. Program `D18F2x9C_x0D0F_0[F,8:0]30_dct[0][BlockRxDqsLock] = 0`.

### 2.9.7.9.4 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 conducted by performing a two dimensional search of the delay settings found in `D18F2x9C_x0000_0[3:0]0[7:5]_dct[0]_mp[1:0]` [DRAM Read DQS Timing] and `D18F2x9C_x0000_0[3:0]0[3:1]_dct[0]_mp[1:0]` [DRAM Write Data Timing].

Training is accomplished on a per channel, per rank, and per lane basis.

For DQS position training, BIOS generates a training pattern using continuous read or write data streams. A 2k-bit-time training pattern is recommended for optimal results. To achieve this, BIOS programs `D18F2x260_dct[0][CmdCount] = 256`, `D18F2x250_dct[0][CmdTgt]=01b`, and `D18F2x25[8,4]_dct[0]` to access two different banks of the same CS. See 2.9.8 [Continuous Pattern Generation].

Prior to DQS position training, BIOS must program `D18F2x210_dct[0]_nbp[3:0][MaxRdLatency]` based on the current greatest value of `D18F2x9C_x0000_00[2A:10]_dct[0]_mp[1:0]`. See 2.9.7.9.5 [Calculating

[MaxRdLatency](#)].

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

- Program [D18F2x9C\\_x0D0F\\_0\[F,8:0\]1F\\_dct\[0\]\\_mp\[1:0\]](#) for all lanes.
  - $RxDqInsDly = 0$ .

For each rank and lane:

1. Select a 64 byte aligned test address.
2. For each write data delay value in [D18F2x9C\\_x0000\\_0\[3:0\]0\[3:1\]\\_dct\[0\]\\_mp\[1:0\]](#) from Wr-DQS to Wr-DQS plus 1 UI, using the Wr-DQS delay value found in 2.9.7.9.1 [\[Write Levelization Training\]](#):
  - Program the write data delay value for the current lane.
  - Write the DRAM training pattern to the test address.
3. For each read DQS delay value in [D18F2x9C\\_x0000\\_0\[3:0\]0\[7:5\]\\_dct\[0\]\\_mp\[1:0\]](#) from 0 to 1 UI:
  - Program the read DQS delay value for the current lane.
  - Read the DRAM training pattern from the test address.
  - Record the result for the current settings as a pass or fail depending if the pattern is read correctly.
  - 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 [D18F2x9C\\_x0000\\_0\[3:0\]0\[7:5\]\\_dct\[0\]\\_mp\[1:0\]](#) with the average value of the smallest and largest delay values in the string of consecutive passing results.
    - If the average value of passing read DQS delay for the lane is negative, then adjust the input receiver DQ delay in [D18F2x9C\\_x0D0F\\_0\[F,8:0\]1F\\_dct\[0\]\\_mp\[1:0\]\[RxDqInsDly\]](#) for the lane as follows:
      - IF ( $RxDqInsDly < 3$ ) THEN increment  $RxDqInsDly$  and repeat step 3 above for all ranks and lanes.
      - ELSE program the read DQS delay for the lane with a value of zero.
      - ENDIF.
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 [D18F2x9C\\_x0000\\_0\[3:0\]0\[3:1\]\\_dct\[0\]\\_mp\[1:0\]](#) with the average value of the smallest and largest delay values in the string of consecutive passing results.

See [Figure 3](#).





### 2.9.7.9.5 Calculating MaxRdLatency

The MaxRdLatency value determines when the memory controller can receive incoming data from the DCTs. 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 `D18F2x210_dct[0]_nbp[3:0][MaxRdLatency]`.

The following steps describe the algorithm used to compute `D18F2x210_dct[0]_nbp[3:0][MaxRdLatency]` used for DRAM training. P, N, and T are used as a temporary placeholders for the incrementally summed value.

1.  $P = N = T = 0$
2. If (`D18F2x9C_x0000_0004_dct[0]_mp[1:0][AddrCmdSetup] = 0` & `D18F2x9C_x0000_0004_dct[0]_mp[1:0][CsOdtSetup] = 0` & `D18F2x9C_x0000_0004_dct[0]_mp[1:0][CkeSetup] = 0`) then  $P = P + 1$  else  $P = P + 2$
3.  $P = P + (8 - \text{D18F2x210\_dct[0]_nbp[3:0][RdPtrInit]})$
4.  $P = P + \text{NumPclks}$ 
  - NumPclks = 6
5.  $P = P + (2 * (\text{D18F2x200\_dct[0]_mp[1:0][Tcl] - 1 \text{ clocks}}))$
6.  $P = P + \text{CEIL}(\text{MAX}(\text{D18F2x9C\_x0000\_00[2A:10]_dct[0]_mp[1:0][DqsRcvEnGrossDelay, DqsRcvEnFineDelay] + \text{D18F2x9C\_x0000\_0[3:0]0[7:5]_dct[0]_mp[1:0][RdDqsTime] PCLKs})) + 1$ 
  - Prior to DQS position training, use maximum value for RdDqsTime.
7. If ( $\text{NclkFreq/MemClkFreq} < 2$ ) then  $P = P + 4.5$  Else  $P = P + 2.5$
8.  $T = T + 1050 \text{ ps}$
9.  $N = (P / (\text{MemClkFreq} * 2) + T) * \text{NclkFreq}$ 
  - See `D18F5x16[C:0][NbDid, NbFid]` and `D18F2x94_dct[0][MemClkFreq]`.
10. `D18F2x210_dct[0]_nbp[3:0][MaxRdLatency] = CEIL(N) + NumNclks`
  - NumNclks = 1

#### 2.9.7.9.5.1 MaxRdLatency Training

After DRAM DQS receiver enable training, BIOS optimizes `D18F2x210_dct[0]_nbp[3:0][MaxRdLatency]` using the following algorithm. For MaxRdLatency training, BIOS generates a training pattern using continuous read or write data streams. See 2.9.8 [Continuous Pattern Generation].

For each channel:

1. Calculate a starting MaxRdLatency delay value by executing the steps in 2.9.7.9.5, excluding steps 4, 7, 8, and 10.
2. Select 32 64-byte aligned test addresses associated with the rank that has the worst case (`D18F2x9C_x0000_00[2A:10]_dct[0]_mp[1:0][DqsRcvEnGrossDelay, DqsRcvEnFineDelay] + D18F2x9C_x0000_0[3:0]0[7:5]_dct[0]_mp[1:0][RdDqsTime]`) register setting.
3. Write the DIMM test addresses with the training pattern.
4. For each MaxRdLatency value incrementing from the value calculated in step 1:
  - A. Program `D18F2x210_dct[0]_nbp[3:0][MaxRdLatency]` with the current value.
  - B. Read the DIMM test addresses.
  - C. Compare the values read against the pattern written.
    - If the pattern is read correctly, go to step 5.
  - D. Program `D18F2x9C_x0000_0050_dct[0]=00000000h`

5. Program `D18F2x210_dct[0]_nbp[3:0][MaxRdLatency] = CEIL(current value + 1 NCLK + 1.5 MEMCLK ENDIF + IF (NclkFreq/MemClkFreq < 2) THEN 1.0 MEMCLK ELSE 0 ENDIF)` .

### 2.9.7.10 DRAM Phy Power Savings

For power savings, BIOS should perform the following actions for each channel:

1. Program `D18F2x88_dct[0][MemClkDis]` to disable unused MEMCLK pins.
2. Program `D18F2x9C_x0D0F_2[F,2:0]30_dct[0][PwrDn]` for unused MEMCLK pairs.
3. Program `D18F2x9C_x0D0F_0[F,8:0]30_dct[0][PwrDn]` to disable the ECC lane if `D18F2x90_dct[0][DimmEccEn]==0`.
4. Program `D18F2x9C_x0000_000C_dct[0][CKETri, ODTTri, ChipSelTri]` to disable unused pins.
5. Program `D18F2x9C_x0D0F_0[F,8:0]13_dct[0]_mp[1:0][DlIDisEarlyU] = 1`.
6. Program `D18F2x9C_x0D0F_0[F,8:0]13_dct[0]_mp[1:0][DlIDisEarlyL] = 1`.
7. `D18F2x9C_x0D0F_812F_dct[0][PARTri] = 1`.
8. `D18F2x9C_x0D0F_812F_dct[0][Add17Tri, Add16Tri] = {1b, 1b}`
9. Program `D18F2x9C_x0D0F_0[F,8:0]10_dct[0]_mp[1:0][EnRxPadStandby] = IF (DdrRate <= 1600) THEN 1 ELSE 0 ENDIF`.
10. Program `D18F2x9C_x0000_000D_dct[0]_mp[1:0]` as follows:
  - If `(DdrRate <= 1600) TxMaxDurDlINoLock = RxMaxDurDlINoLock = 8h`  
else `TxMaxDurDlINoLock = RxMaxDurDlINoLock = 7h`.
  - `TxCPUpdPeriod = RxCPUpdPeriod = 011b`.
  - `TxDLLWakeupTime = RxDLLWakeupTime = 11b`.
11. Program `D18F2x9C_x0D0F_0[F,8:0]1C_dct[0]_mp[1:0]` as follows:
  - If `D18F2x90_dct[0][DimmEccEn]` Numlanes = 9 Else Numlanes = 8.
  - If `(DdrRate >= 1866) DllWakeTime = 1` Else `DllWakeTime = 0`.
  - Let `MaxRxStggrDly = ((Tcl-1)*2) + MIN(DqsRcvEnGrossDelay for all byte lanes) - 6`.
    - See `D18F2x9C_x0000_00[2A:10]_dct[0]_mp[1:0]`.
  - Let `dRxStggrDly = (MaxRxStggrDly - DllWakeTime) / (Numlanes - 1)`.
  - For each byte lane in the ordered sequence below, program `RxDllStggrDly[5:0]` = an increasing value, starting with 0 for the first byte lane in the sequence and increasing at a rate of `dRxStggrDly` for each subsequent byte lane.
    - Sequence: If `D18F2x90_dct[0][DimmEccEn]` {8, 4, 3, 5, 2, 6, 1, 7, 0} else {4, 3, 5, 2, 6, 1, 7, 0}.
  - Let `MaxTxStggrDly = MIN(((Tcwl-1)*2) + MIN(WrDqsGrossDly for all byte lanes) - WrDqDqsEarly - 4, MaxRxStggrDly)`
    - See `D18F2x9C_x0000_00[4A:30]_dct[0]_mp[1:0]`.
    - See `D18F2x20C_dct[0]_mp[1:0]`.
  - Let `dTxStggrDly = (MaxTxStggrDly - DllWakeTime) / (Numlanes - 1)`.
  - For each byte lane in the ordered sequence below, program `TxDllStggrDly[5:0]` = an increasing integer value, starting with 0 for the first byte lane in the sequence and increasing at a rate of `dTxStggrDly` for each subsequent byte lane.
    - Sequence: If `D18F2x90_dct[0][DimmEccEn]` {0, 7, 1, 6, 2, 5, 3, 4, 8} else {0, 7, 1, 6, 2, 5, 3, 4}.
  - `RxDllStggrEn = TxDllStggrEn = 1`.
12. Program `D18F2x248_dct[0]_mp[1:0]` and then `D18F2x9C_x0D0F_0[F,8:0]13_dct[0]_mp[1:0]` as follows:
  - For M0 & M1 context program `RxChMntClkEn=RxSsbMntClkEn=0`.
13. Program `D18F2x9C_x0D0F_0[F,8:0]30_dct[0][TxPclkGateEn, PchgPdPclkGateEn, DataCtlPipePclkGateEn] = {1, 1, 1}`.
14. Program `D18F2x9C_x0D0F_0[F,8:0]04_dct[0]_mp[1:0][TriDM] = IF (DataMaskMbType == 0) THEN 1 ELSE 0 ENDIF`.

## 2.9.8 Continuous Pattern Generation

DRAM training relies on the ability to generate a string of continuous reads or writes between the processor and DRAM, such that worst case electrical interactions can be created. This section describes how these continuous strings of accesses may be generated.

### 2.9.8.1 DCT Training Pattern Generation

DCT training pattern generation uses pattern generators in the DCT to generate controlled read and write traffic streams. During write pattern generation, data values based off of a deterministic pattern are burst to the DRAM interface. Conversely for reads, data bursts from the DRAM interface are compared against expected data values on a per nibble basis.

Two address modes are available for DRAM training pattern generation, as configured by [D18F2x250\\_dct\[0\]\[CmdTgt\]](#). For generating a stream of reads or writes to the same rank, address target A mode is used. To generate a stream of accesses to up to two different ranks, address target A and B mode is used.

An overview of the BIOS sequence to generate training patterns is as follows:

- Configure the DCT for pattern generation. See [2.9.7.7 \[DCT Training Specific Configuration\]](#).
- Ensure DIMMs are configured to support 8-beat bursts (BL8 or dynamic burst length on the fly).
- Wait for [D18F2x250\\_dct\[0\]\[CmdSendInProg\] = 0](#).
- Program [D18F2x250\\_dct\[0\]\[CmdTestEnable\] = 1](#).
- Send activate commands as appropriate. See [2.9.8.1.1 \[Activate and Precharge Command Generation\]](#).
- Send read or write commands as desired. See [2.9.8.1.2 \[Read and Write Command Generation\]](#).
- Send precharge commands as appropriate. See [2.9.8.1.1 \[Activate and Precharge Command Generation\]](#).
- Program [D18F2x250\\_dct\[0\]\[CmdTestEnable\] = 0](#).

#### 2.9.8.1.1 Activate and Precharge Command Generation

Prior to sending read or write commands, BIOS must send an activate command to a row in a particular bank of the DRAM devices for access. To send an activate command, BIOS performs the following steps:

- Program [D18F2x28C\\_dct\[0\]](#) to the desired address as follows:
  - [CmdChipSelect = CS\[7:0\]](#).
  - [CmdBank = BA\[2:0\]](#).
  - [CmdAddress = A\[17:0\]](#).
- Program [D18F2x28C\\_dct\[0\]\[SendActCmd\] = 1](#).
- Wait until [D18F2x28C\\_dct\[0\]\[SendActCmd\] = 0](#).
- Wait 75 MEMCLKs.

After completing its accesses, BIOS must deactivate open rows in the DRAM devices. To send a precharge or precharge all command to deactivate open rows in a bank or in all banks, BIOS performs the following steps:

- Wait 25 MEMCLKs.
- Program [D18F2x28C\\_dct\[0\]](#) to the desired address as follows:
  - [CmdChipSelect = CS\[7:0\]](#).
  - Precharge all command:
    - [CmdAddress\[10\] = 1](#).
  - Precharge command:
    - [CmdAddress\[10\] = 0](#).
    - [CmdBank = BA\[2:0\]](#).
- Program [D18F2x28C\\_dct\[0\]\[SendPchgCmd\] = 1](#).



- Wait until `D18F2x28C_dct[0][SendPchgCmd] = 0`.
- Wait 25 MEMCLKs.

### 2.9.8.1.2 Read and Write Command Generation

BIOS performs the following steps for read pattern generation:

- Program `D18F2x27C_dct[0]`, `D18F2x278_dct[0]`, and `D18F2x274_dct[0]` with the data comparison masks for bit lanes of interest.
- Program `D18F2x270_dct[0][DataPrbsSeed]` the seed for the desired PRBS.
- Program `D18F2x260_dct[0][CmdCount]` equal to the number of cache line commands.
- Program `D18F2x25C_dct[0][BubbleCnt, CmdStreamLen]`. See 2.9.8.1.5 [BubbleCnt and CmdStream-Len Programming].
- Program `D18F2x25[8,4]_dct[0]` to the initial address.
- Program `D18F2x250_dct[0]` as follows:
  - ResetAllErr and StopOnErr as desired. See 2.9.8.1.4 [Data Comparison].
  - CmdTgt corresponding to `D18F2x25[8,4]_dct[0]`.
  - CmdType = 000b.
  - SendCmd = 1.
- If `D18F2x260_dct[0][CmdCount] != 0` then  
Wait for `D18F2x250_dct[0][TestStatus] = 1` and `D18F2x250_dct[0][CmdSendInProg] = 0`.  
Else  
Wait the desired amount of time.  
Program `D18F2x260_dct[0][CmdCount] = 1`.  
Wait for `D18F2x250_dct[0][TestStatus] = 1` and `D18F2x250_dct[0][CmdSendInProg] = 0`.
- Program `D18F2x250_dct[0][SendCmd] = 0`.
- Read `D18F2x264_dct[0]`, `D18F2x268_dct[0]`, and `D18F2x26C_dct[0]` if applicable.

BIOS performs the following steps for write pattern generation:

- Program `D18F2x270_dct[0][DataPrbsSeed]` the seed for the desired PRBS.
- Program `D18F2x260_dct[0][CmdCount]` equal to the number of cache line commands desired.
- Program `D18F2x25C_dct[0][BubbleCnt, CmdStreamLen]`. See 2.9.8.1.5 [BubbleCnt and CmdStream-Len Programming].
- Program `D18F2x25[8,4]_dct[0]` to the initial address.
- Program `D18F2x250_dct[0]` as follows:
  - CmdTgt corresponding to `D18F2x25[8,4]_dct[0]`.
  - CmdType = 001b.
  - SendCmd = 1.
- If `D18F2x260_dct[0][CmdCount] != 0` then  
Wait for `D18F2x250_dct[0][TestStatus] = 1` and `D18F2x250_dct[0][CmdSendInProg] = 0`.  
Else  
Wait the desired amount of time.  
Program `D18F2x260_dct[0][CmdCount] = 1`.  
Wait for `D18F2x250_dct[0][TestStatus] = 1` and `D18F2x250_dct[0][CmdSendInProg] = 0`.
- Program `D18F2x250_dct[0][SendCmd] = 0`.

BIOS combines the two sets of steps listed above for alternating write and read pattern generation.

### 2.9.8.1.3 Configurable Data Patterns

In addition to PRBS mode, `D18F2x250_dct[0][DataPatGenSel]` and `D18F2x2[B4,B0,AC,A8]_dct[0]` allow

configurable data pattern generation.

**Table 53. Configurable Data Pattern Example with 1 Address Target**

DQ/ Beat	Write TgtA Cmd 0							Write TgtA Cmd 1							Write TgtA Cmd 2							...				
	Beat 0	Beat 1	Beat 2	Beat 3	Beat 4	Beat 5	Beat 6	Beat 7	Beat 8	Beat 9	Beat 10	Beat 11	Beat 12	Beat 13	Beat 14	Beat 15	Beat 16	Beat 17	Beat 18	Beat 19	Beat 20	Beat 21	Beat 22	Beat 23	...	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
1	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0		
2	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0		
3	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0		
4	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0		
5	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0		
6	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0		
7	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0		
8	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1		
9	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1		
10	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1		
11	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1		
12	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1		
13	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1		
14	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1		
15	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
...																										
ECC0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
ECC1	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0		
ECC2	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0		
ECC3	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0		
ECC4	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0		
ECC5	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0		
ECC6	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0		
ECC7	1	1	1	0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	0	1	1	1	1		
1. D18F2x250_dct[0][DataPatGenSel] = 10b D18F2x2[B4,B0,AC,A8]_dct[0] = {FFEE_DDCCh, BBAA_9988h, 7766_5544h, 3322_1100h}																										

**Table 54. Configurable Data Pattern Circular Shift Example with 1 Address Target**

DQ/ Beat	Write TgtA Cmd 0							Write TgtA Cmd 1							Write TgtA Cmd 2							...		
	Beat 0	Beat 1	Beat 2	Beat 3	Beat 4	Beat 5	Beat 6	Beat 7	Beat 8	Beat 9	Beat 10	Beat 11	Beat 12	Beat 13	Beat 14	Beat 15	Beat 16	Beat 17	Beat 18	Beat 19	Beat 20	Beat 21	Beat 22	Beat 23
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	1	1	1	0	1	1	1	
1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	
2	0	1	0	0	0	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	
3	1	1	0	0	1	1	0	0	1	0	0	0	1	0	0	1	0	0	0	1	0	0	0	
4	0	0	1	0	0	0	1	0	1	1	0	0	1	1	0	0	0	1	0	0	0	1	0	
5	1	0	1	0	1	0	1	0	0	0	1	0	0	0	1	0	1	1	0	0	1	1	0	
6	0	1	1	0	0	1	1	0	1	0	1	0	1	0	1	0	0	0	1	0	0	0	1	
7	1	1	1	0	1	1	1	0	0	1	1	0	0	1	1	0	1	0	1	0	1	0	1	
8	0	0	0	1	0	0	0	1	1	1	1	0	1	1	1	0	0	1	1	0	0	1	1	
9	1	0	0	1	1	0	0	1	0	0	0	1	0	0	0	1	1	1	1	0	1	1	1	
10	0	1	0	1	0	1	0	1	1	0	0	1	1	0	0	1	0	0	0	1	0	0	0	
11	1	1	0	1	1	1	0	1	0	1	0	1	0	1	0	1	0	0	1	1	0	0	1	
12	0	0	1	1	0	0	1	1	1	1	0	1	1	1	0	1	0	1	0	1	0	1	0	
13	1	0	1	1	1	0	1	1	0	0	1	1	0	0	1	1	1	0	1	1	1	0	1	
14	0	1	1	1	0	1	1	1	1	0	1	1	1	0	1	1	0	0	1	1	0	0	1	
15	1	1	1	1	1	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	
...																								
ECC0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0	1	1	1	0	1	1	1	
ECC1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	
ECC2	0	1	0	0	0	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	
ECC3	1	1	0	0	1	1	0	0	0	1	0	0	0	1	0	0	1	0	0	0	1	0	0	
ECC4	0	0	1	0	0	0	1	0	1	1	0	0	1	1	0	0	0	1	0	0	0	1	0	
ECC5	1	0	1	0	1	0	1	0	0	0	1	0	0	0	1	0	1	1	0	0	1	1	0	
ECC6	0	1	1	0	0	1	1	0	1	0	1	0	1	0	1	0	0	0	1	0	0	0	1	
ECC7	1	1	1	0	1	1	1	1	0	1	1	0	0	1	1	0	1	0	1	0	1	0	1	

1. D18F2x250\_dct[0][DataPatGenSel] = 11b  
D18F2x2[B4,B0,AC,A8]\_dct[0] = {FFEE\_DDCCh, BBAA\_9988h, 7766\_5544h, 3322\_1100h}

**2.9.8.1.3.1 Static Data Pattern Override**

Software is also able to use the bits of D18F2x280\_dct[0], D18F2x284\_dct[0], and D18F2x288\_dct[0] to override the DCT pattern generator on a DQ basis. This can only be used in the D18F2x250\_dct[0][DataPatGenSel] = 10b mode. Software programs the following to enable the mode:

- Program D18F2x288\_dct[0][PatOvrVal] to the data value desired.
- Program D18F2x280\_dct[0][DQPatOvrEn[31:0]], D18F2x284\_dct[0][DQPatOvrEn[63:32]], and D18F2x288\_dct[0][EccPatOvrEn] to enable the override on the desired bit lanes.

**Table 55. Data Pattern Override Example with 1 Address Target**

DQ/ Beat	Write TgtA Cmd 0								Write TgtA Cmd 1								Write TgtA Cmd 2								...
	Beat 0	Beat 1	Beat 2	Beat 3	Beat 4	Beat 5	Beat 6	Beat 7	Beat 8	Beat 9	Beat 10	Beat 11	Beat 12	Beat 13	Beat 14	Beat 15	Beat 16	Beat 17	Beat 18	Beat 19	Beat 20	Beat 21	Beat 22	Beat 23	...
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
1	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	
2	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	
3	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	
4	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	
5	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	
6	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
8	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	
9	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	
10	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	
11	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	
12	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	
13	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	
14	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	
15	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
...																									
ECC0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
ECC1	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	
ECC2	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	
ECC3	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	
ECC4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
ECC5	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	
ECC6	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	
ECC7	1	1	1	0	1	1	1	1	1	0	1	1	1	1	1	1	1	1	0	1	1	1	1		

1. D18F2x250\_dct[0][DataPatGenSel] = 10b  
 D18F2x2[B4,B0,AC,A8]\_dct[0] = {FFEE\_DDCCh, BBAA\_9988h, 7766\_5544h, 3322\_1100h}  
 D18F2x288\_dct[0][PatOvrVal] = 0.  
 D18F2x280\_dct[0][DQPatOvrEn[31:0]] = 0x0000\_0080.  
 D18F2x284\_dct[0][DQPatOvrEn[63:32]] = 0x0000\_0000.  
 D18F2x288\_dct[0][EccPatOvrEn] = 0x10.

### 2.9.8.1.3.2 Xor Data Pattern Override

Software is also able to use the bits of [D18F2x288\\_dct\[0\]\[XorPatOvr\]](#) to override the DCT pattern generator with the same XOR function for all byte lanes. This feature can only be used with select [D18F2x250\\_dct\[0\]\[DataPatGenSel\]](#) PRBS modes. Software programs the following to enable the mode:

- Program `D18F2x288_dct[0][XorPatOvr]` to the data value of the XOR function desired.
  - Output data = IF (DQPatOvrEn) THEN PatOvrVal ELSE (DataPatGen ^ XorPatOvr) ENDIF.

### 2.9.8.1.4 Data Comparison

The DCT compares the incoming read data against the expected pattern sequence during pattern generation. BIOS may choose to continue command generation and accumulate errors or stop command generation on the first error occurrence by programming `D18F2x250_dct[0][StopOnErr]`.

Error information is reported via `D18F2x264_dct[0]`, `D18F2x268_dct[0]`, `D18F2x26C_dct[0]`, `D18F2x294_dct[0]`, `D18F2x298_dct[0]` and `D18F2x29C_dct[0]` Error information can be masked on per-bit basis by programming `D18F2x274_dct[0]`, `D18F2x278_dct[0]`, and `D18F2x27C_dct[0]`.

BIOS resets the error information by programming `D18F2x250_dct[0][ResetAllErr]=1`.

Error information is only valid in certain modes of `D18F2x250_dct[0][CmdType, CmdTgt]` and `D18F2x260_dct[0][CmdCount]` and when using 64 byte aligned addresses in `D18F2x25[8,4]_dct[0][TgtAddress]`. Some modes require a series of writes to setup a DRAM data pattern. See Table 56.

**Table 56. Command Generation and Data Comparison**

Commands	CmdType	Cmd Tgt	Maximum CmdCount <sup>4</sup>
Read	000b	00b <sup>1</sup>	128
		01b <sup>1</sup>	256 <sup>2</sup>
Write-Read	010b	00b	Infinite
		01b <sup>3</sup>	256 <sup>2</sup>

- Requires setup writes to store a data pattern in DRAM. The write commands are generated using the same CmdTgt, CmdCount, and DataPrbsSeed settings.
- `D18F2x254[TgtAddress] != D18F2x258[TgtAddress]`.
- Requires setup writes to store a data pattern in DRAM. The write commands are generated programming `D18F2x254[TgtAddress]` to the intended Target B, `CmdTgt=00b`, `CmdCount` to 1/2 of the intended command count, and the same `DataPrbsSeed` setting.
- `D18F2x250_dct[0][LsfrRollOver]=0`. The maximum `CmdCount` is infinite for all modes listed if `D18F2x250_dct[0][LsfrRollOver]=1`.

#### 2.9.8.1.4.1 Activate and Precharge Traffic Generation

The DCT generates ACT and PRE traffic in the available command bandwidth during a WR or RD sequence generated by `D18F2x250_dct[0][SendCmd]` when `D18F2x250_dct[0][ActPchgGenEn]=1`. This 16 command sequence repeats until the WR or RD initiated by `SendCmd` hits a stopping condition.

Software is able to adjust the command spacing and address sequence by programming `D18F2x28C_dct[0][CmdChipSelect, CmdAddress[17:4]]`, `D18F2x2B8_dct[0][ActPchgSeq, ActPchgCmdMin]`, and `D18F2x2[C0,BC]_dct[0]`.

The banks specified by `D18F2x2[CO,BC]_dct[0]` must be in the idle state and must not conflict with the banks used by the SendCmd WR or RD traffic stream. Software is responsible for the ACT/PRE protocol of the WR or RD targets initiated by SendCmd. On a stopping condition for the WR or RD traffic stream, software is responsible for returning all of the possibly open banks to the idle state.

- Example configuration for DRAM Training:
- `D18F2x250_dct[0][ActPchgGenEn, CmdTgt, CmdType] = { 1b, 1b, 0b}`. // Tgt A&B, Reads
- `D18F2x28C_dct[0][CmdChipSelect] = CS of TgtA`
- `D18F2x28C_dct[0][CmdAddress[17:4]] = desired upper row address bits.`
  - Hardware places the sequence number in row address [3:0].
- `D18F2x2B8_dct[0][ActPchgSeq, ActPchgCmdMin]= {0F0Fh, Fh}`.
- `D18F2x2BC_dct[0] = 5432_ 5432h`; // Banks 2,3,4,5
- `D18F2x2C0_dct[0] = 7632_ 7632h`; // Banks 2,3,6,7

### 2.9.8.1.5 BubbleCnt and CmdStreamLen Programming

BIOS programs `D18F2x25C_dct[0][BubbleCnt2, BubbleCnt, CmdStreamLen]` to ensure proper channel command spacing in command generation mode.

For continuous pattern generation it is expected that `BubbleCnt = 0`. In other modes, BIOS programs `BubbleCnt`, `BubbleCnt2`, and `CmdStreamLen` greater than or equal to the relevant DRAM timing parameters shown below to prevent contention on the DRAM bus. In all cases, if the minimum `BubbleCnt > 0` or `CmdType = 010b` then BIOS programs `CmdStreamLen = 1`.

**Table 57. DDR3 Command Generation and BubbleCnt Programming**

Commands	CmdType	CmdTgt	BubbleCnt	BubbleCnt2
Read-Read same CS	000b	0xb	<code>D18F2x218_dct[0]_mp[1:0][TrdrdSdSc] - 1</code> ; Exclude banned spacing: <code>D18F2x218_dct[0]_mp[1:0][TrdrdBan]</code>	xh
Write-Write same CS	001b	0xb	<code>D18F2x214_dct[0]_mp[1:0][TwrwrSdSc] - 1</code>	xh
Write-Read same CS	010b	00b	<code>D18F2x20C_dct[0]_mp[1:0][Twtr] + D18F2x20C_dct[0]_mp[1:0][Tcwl] + 4 - 1</code>	<code>D18F2x21C_dct[0]_mp[1:0][TrwtTO] - 1</code>
Read-Read different CS	000b	01b	<code>D18F2x218_dct[0]_mp[1:0][TrdrdSdDc] - 1</code> ; Exclude banned spacing: <code>D18F2x218_dct[0]_mp[1:0][TrdrdBan]</code>	xh
Write-Write different CS	001b	01b	<code>D18F2x214_dct[0]_mp[1:0][TwrwrSdDc] - 1</code>	xh
Write-Read different CS	010b	01b	<code>D18F2x218_dct[0]_mp[1:0][Twrrd] - 1</code>	<code>D18F2x21C_dct[0]_mp[1:0][TrwtTO] - 1</code>

**Table 57. DDR3 Command Generation and BubbleCnt Programming**

Commands	CmdType	CmdTgt	BubbleCnt	BubbleCnt2
Read-Read different DIMM	000b	01b	D18F2x218_dct[0]_mp[1:0][TrdrdDd] - 1; Exclude banned spacing: D18F2x218_dct[0]_mp[1:0][TrdrdBan]	xh
Write-Write different DIMM	001b	01b	D18F2x214_dct[0]_mp[1:0][TwrwrDd] - 1	xh
Write-Read different DIMM	010b	01b	D18F2x218_dct[0]_mp[1:0][Twrdrd] - 1	D18F2x21C_dct[0]_mp[1:0][TrwtTO] - 1

### 2.9.9 Memory Interleaving Modes

The processor supports the following memory interleaving modes:

- Chip select: interleaves the physical address space over multiple DIMM ranks on a channel, as opposed to each DIMM owning single consecutive address spaces. See 2.9.9.1 [Chip Select Interleaving].

**Table 58. Recommended Interleave Configurations**

Interleaving Mode	Enabled	Disabled
Chip Select Interleaving	Number of chip selects installed on the channel is a power of two.	Requirements not satisfied.

#### 2.9.9.1 Chip Select Interleaving

The chip select memory interleaving mode has the following requirements:

- The number of chip selects interleaved is a power of two.
- The chip selects are the same size and type.

A BIOS algorithm for programming D18F2x[5C:40]\_dct[0] [DRAM CS Base Address] and D18F2x[6C:60]\_dct[0] [DRAM CS Mask] 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[38:27] bits with the BaseAddr[21:11] bits as defined in Table 59.
3. For each enabled chip select, swap the corresponding AddrMask[38:27] bits with the AddrMask[21:11] bits as defined in Table 59.

**Table 59. DDR3 Swapped Normalized Address Lines for CS Interleaving**

Condition				Swapped Base Address and Address Mask bits	
DIMM Address Map <sup>1</sup>	CS Size	BankSwap <sup>2</sup>	DctSelBankSwap <sup>2</sup>	4 way CS interleaving	2 way CS interleaving
0001b	256MB	0	-	[29:28] and [17:16]	[28] and [16]
		1	1	[29:28] and [12:11]	[28] and [11]
		1	0	[29:28] and [13:12]	[28] and [12]
0010b	512MB	0	-	[30:29] and [17:16]	[29] and [16]
		1	1	[30:29] and [12:11]	[29] and [11]
		1	0	[30:29] and [13:12]	[29] and [12]
0101b	1GB	0	-	[31:30] and [17:16]	[30] and [16]
		1	1	[31:30] and [12:11]	[30] and [11]
		1	0	[31:30] and [13:12]	[30] and [12]
0111b	2GB	0	-	[32:31] and [17:16]	[31] and [16]
		1	1	[32:31] and [12:11]	[31] and [11]
		1	0	[32:31] and [13:12]	[31] and [12]
1010b	4GB	0	-	[33:32] and [17:16]	[32] and [16]
		1	1	[33:32] and [12:11]	[32] and [11]
		1	0	[33:32] and [13:12]	[32] and [12]
1011b	8GB	0	-	[34:33] and [18:17]	[33] and [17]
		1	1	[34:33] and [12:11]	[33] and [11]
		1	0	[34:33] and [13:12]	[33] and [12]

1. See [D18F2x80\\_dct\[0\]](#) [DRAM Bank Address Mapping].  
 2. See [D18F2xA8\\_dct\[0\]](#)[BankSwap] and [D18F2x114](#)[DctSel-BankSwap].

**2.9.10 Memory Hoisting**

Memory hoisting reclaims the otherwise inaccessible DRAM that would naturally reside in memory regions used by MMIO. When memory hoisting is configured by BIOS, DRAM physical addresses are repositioned above the 4 GB address level in the address map. In operation, the physical addresses are remapped in hardware to the normalized addresses used by a DCT.

The region of DRAM that is hoisted is defined to be from [D18F1xF0](#)[DramHoleBase] to the 4 GB level. Hoisting is enabled by programming [D18F1xF0](#) [DRAM Hole Address] and configuring the DCTs per the equations in this section.

DramHoleSize is defined in order to simplify the following equations in this section and is calculated as follows:

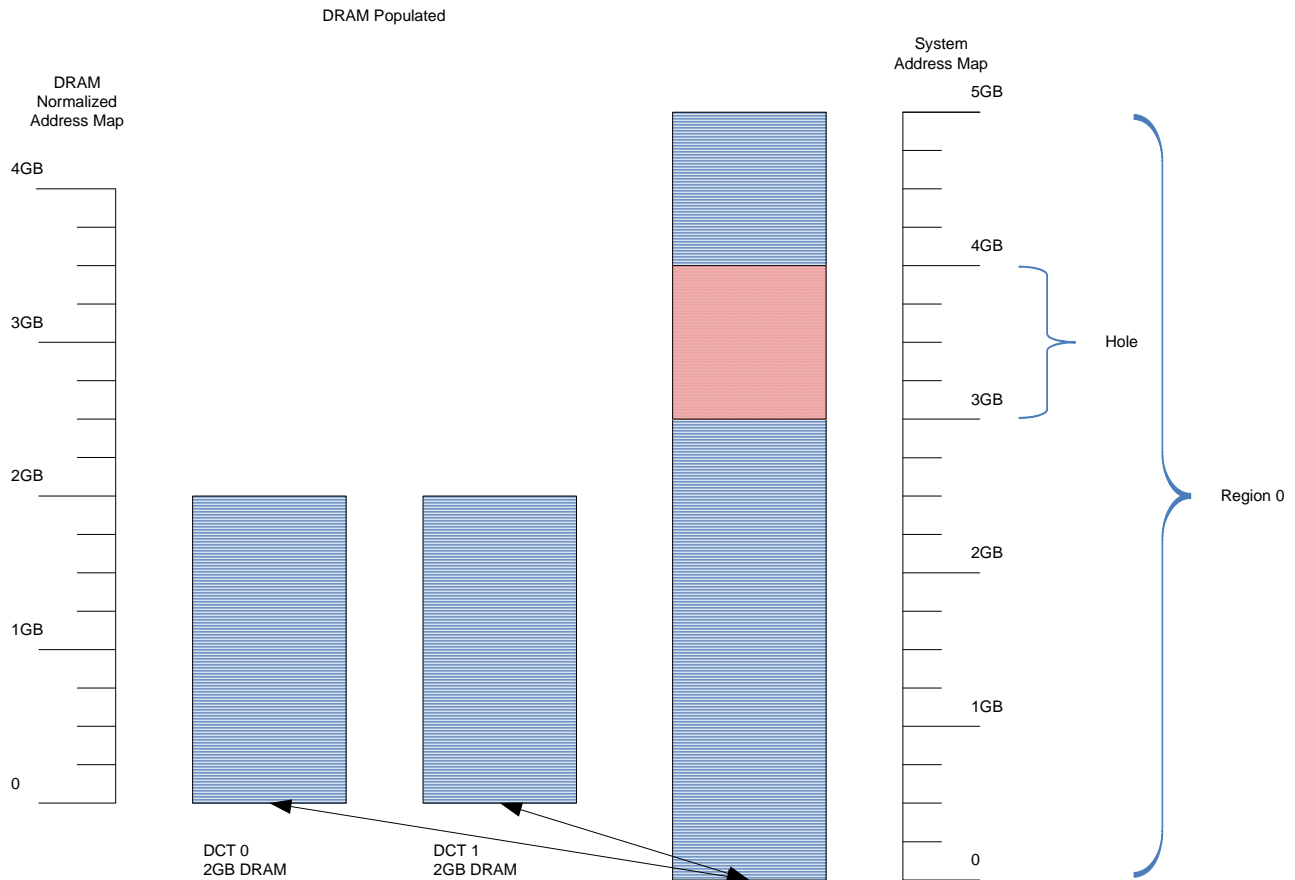


- Define the DRAM hole region as  $\text{DramHoleSize}[31:24] = 100\text{h} - \text{D18F1xF0}[\text{DramHoleBase}[31:24]]$ .

### 2.9.10.1 DramHoleOffset Programming

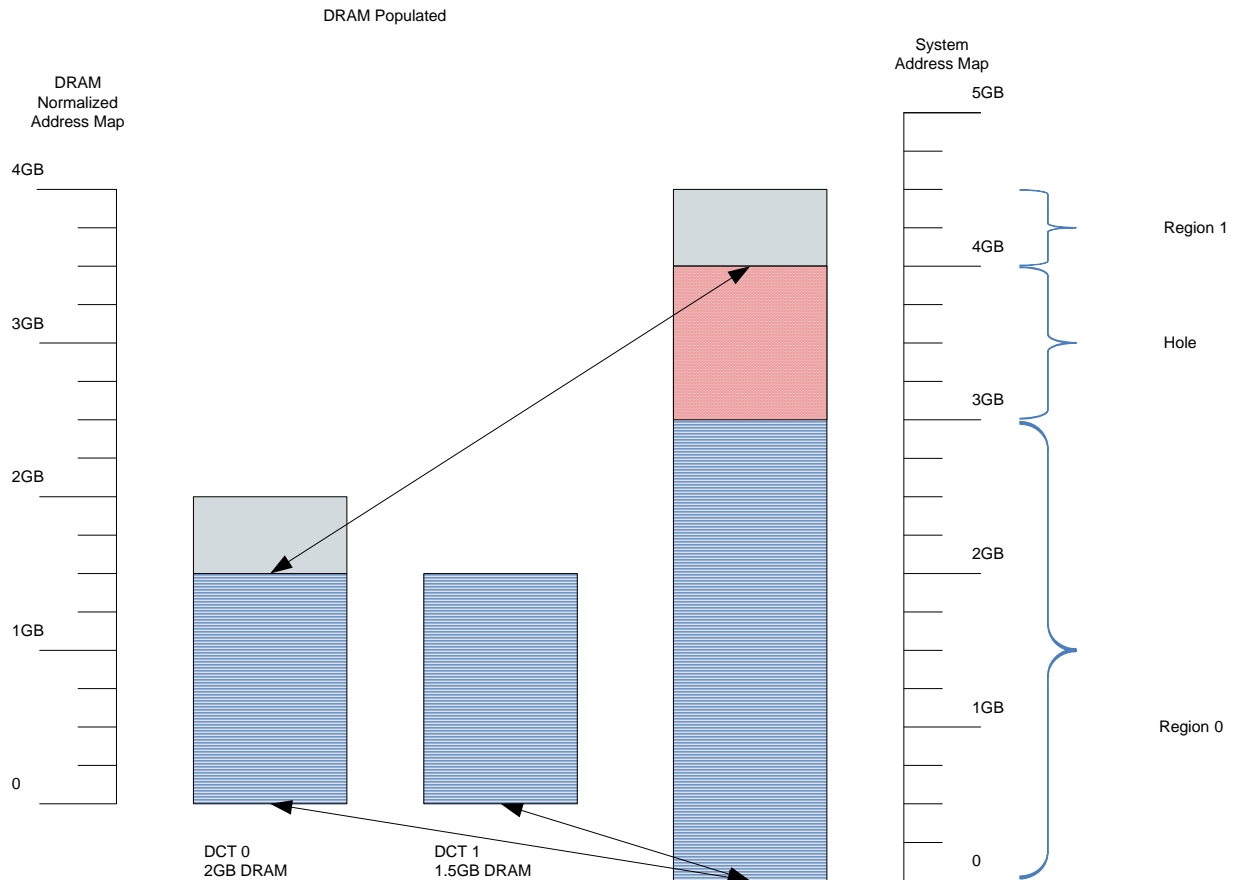
$\text{D18F1xF0}[\text{DramHoleOffset}]$  is programmed to account for the addresses from  $\text{D18F1xF0}[\text{DramHoleBase}]$  to 4 GB when it falls inside of a  $\text{D18F1x2}[1,0][8,0][\text{DctBaseAddr}]$  and  $\text{D18F1x2}[1,0][\text{C},4][\text{DctLimitAddr}]$  region. See Figure 5 as an example memory population.

- Program  $\text{D18F1xF0}[\text{DramHoleOffset}[31:23]] = \{\text{DramHoleSize}[31:24], 0\text{b}\} + \{\text{DctBaseAddr}[31:27], 0000\text{b}\}$ ;



**Figure 5: Memory Configuration with Memory Hole inside of Region**

$\text{D18F1xF0}[\text{DramHoleOffset}]$  is unused when the memory hole falls outside of a region. Figure 6 shows an example memory population which uses two memory regions. Region 1 is configured to begin above the memory hole.



**Figure 6: Memory Configuration with Memory Hole outside of Region**

**2.9.11 DRAM CC6/PC6 Storage**

DRAM is used to hold the state information of cores entering the CC6 power management state. As part of the system setup if CC6 or 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.

**2.9.11.1 Fixed Storage**

The size of each special DRAM storage region is defined to be a fixed 16MB. BIOS configures the storage region at the top of the DRAM range, adjusts `D18F1x[7:4][C,4][DramLimit]` and the processor top of DRAM specified by `MSRC001_001A[TOM]` or `MSRC001_001D[TOM2]` downward accordingly. See [Table 60](#).

After finalizing the system DRAM configuration, BIOS must set `D18F2x118[LockDramCfg] = 1` to enable the hardware protection.

**Table 60. Example storage region configuration**

Node	DRAM Populated	D18F1x[17C:140,7C:40] [DramBase, DramLimit]	CC6 DRAM Range	D18F4x128 [CoreStateSa veDestNode]	D18F1x120[DramBaseAddr], D18F1x124[DramLimitAddr]
0	256 MB	0 MB, 240 MB - 1	240 MB, 256 MB - 1	0	0 MB, 256 MB - 1

### 2.9.12 DRAM On DIMM Thermal Management and Power Capping

Each DCT can throttle commands based on the state of the channel EVENT\_L pin or when D18F2xA4[BwCapEn]=1. The EVENT\_L pin is used for thermal management while D18F2xA4[BwCapEn] limits memory power independent of the thermal management solution.

The recommended BIOS configuration for the EVENT\_L pin is as follows:

- BIOS may enable command throttling on a DRAM controller if the platform supports the EVENT\_L pin by programming D18F2xA4[ODTSEn] = 1.
  - The recommended usage is for this pin to be connected to one or more JEDEC defined on DIMM temperature sensors. The DIMM SPD ROM indicates on DIMM temperature sensor support.
  - BIOS configures the temperature sensor(s) to assert 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[CmdThrottleMode] with the throttling mode to employ when the trip point has been exceeded.
  - The hardware enforces a refresh rate of 3.9 us while EVENT\_L is asserted.
- BIOS configures D18F2x8C\_dct[0][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).
  - If all DIMMs support the normal temperature range, or if normal and extended temperature range DIMMs are mixed, BIOS programs D18F2x8C\_dct[0][Tref] to 7.8 us and D18F2xA4[ODTSEn] = 1. BIOS configures the temperature sensor trip point for all DIMMs according to the 85 °C case temperature specification.
  - If all DIMMs support the extended temperature range, BIOS has two options:
    - a. Follow the recommendation for normal temperature range DIMMs.
    - b. Program D18F2x8C\_dct[0][Tref] = 3.9 us and configure the temperature sensor trip point for all DIMMs according to the 95 °C case temperature specification.
- At startup, the 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 DCT latched status of the EVENT\_L pin for can be read by system software in D18F2xAC [DRAM Controller Temperature Status].

The relationship between the DRAM case temperature, trip point, and EVENT\_L pin sampling interval is outlined as follows:

- The trip point for each DIMM is ordinarily configured to the case temperature specification minus a guardband temperature for the DIMM.
- The temperature guardband is vendor defined and is used to account for sensor inaccuracy, EVENT\_L pin sample interval, and platform thermal design.
- The sampling interval is vendor defined. It is expected to be approximately 1 second.

BIOS may enable bandwidth capping on a DRAM controller by setting `D18F2xA4[BwCapEn] = 1` and programming `D18F2xA4[BwCapCmdThrottleMode]` with the throttling mode to employ. The DCT will employ the larger of the two throttling percentages as specified by `D18F2xA4[BwCapCmdThrottleMode]` and `D18F2xA4[CmdThrottleMode]` if the `EVENT_L` pin is asserted when both `D18F2xA4[BwCapEn] = 1` and `D18F2xA4[ODTSEn] = 1`.

## 2.10 Thermal Functions

Thermal functions 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 it from damage.

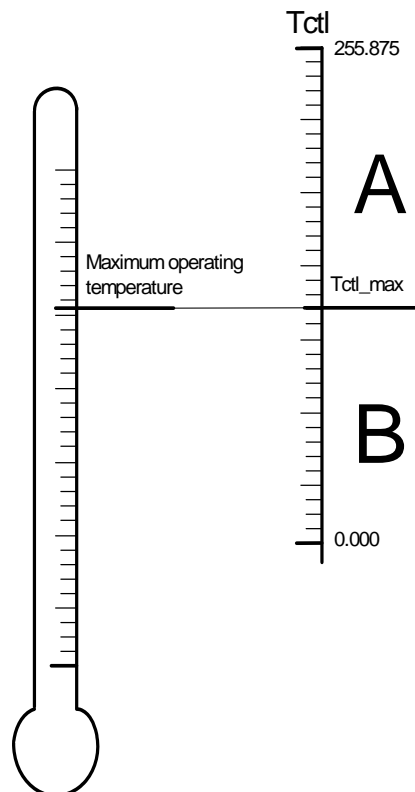
The processor thermal-related circuitry includes (1) the temperature calculation circuit (TCC) for determining the temperature of the processor and (2) logic that uses the temperature from the TCC.

### 2.10.1 The Tctl Temperature Scale

Tctl is a processor temperature control value used for processor thermal management. Tctl is accessible through [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  $Tctl = Tctl\_max$  to 255.875: the temperature of the part is  $[Tctl - Tctl\_max]$  over the maximum operating temperature. The processor may take corrective actions that affect performance, such as HTC, to support the return to Tctl range B.

B: For  $Tctl = 0$  to  $Tctl\_max - 0.125$ : the temperature of the part is  $[Tctl\_max - Tctl]$  under the maximum operating temperature.



**Figure 7: Tctl scale**

### 2.10.2 Temperature Slew Rate Control

The temperature slew rate controls in [D18F3xA4](#) are used to filter the processor temperature provided in [D18F3xA4\[CurTmp\]](#). Separate controls are provided for increasing and decreasing temperatures. The latest measured temperature is referred to as Tctlm below.

If downward slew control is enabled ([D18F3xA4\[TmpSlewDnEn\]](#)), Tctl is not updated down unless Tctlm remains below Tctl for a time specified by [D18F3xA4\[PerStepTimeDn\]](#). If at any point before the timer expires Tctlm equals or exceeds Tctl, then the timer resets and Tctl is not updated. If the timer expires, then Tctl is reduced by 0.125. If downward slew control is disabled, then if Tctlm is less than Tctl, Tctl is immediately updated to Tctlm.

The upward slew control works similar to downward slew control except that if Tctlm exceeds Tctl by a value defined by [D18F3xA4\[TmpMaxDiffUp\]](#) then Tctl is immediately updated to Tctlm. Otherwise, Tctlm must remain above Tctl for time specified by [D18F3xA4\[PerStepTimeUp\]](#) before Tctl is incremented by 0.125.

### 2.10.3 Temperature-Driven Logic

The temperature calculated by the TCC is used by HTC, THERMTRIP, and PROCHOT\_L.

#### 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, software should not change the following: All [D18F3x64](#) fields (except for HtcActSts and HtcEn), [MSRC001\\_001F\[DisProcHotPin\]](#). Any change to the previous list of fields when 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 with a minimum period of 2 ms.

While in the HTC-active state, the following power reduction actions are taken:

- CPU cores are limited to a P-state (specified by [D18F3x64\[HtcPstateLimit\]](#)); see [2.5.3 \[CPU Power Management\]](#)
- The GPU may be placed in a low power state based on the state of [D18F5x178\[ProcHotToGnbEn\]](#)

The processor enters the HTC-active state if all of the following conditions are true:

- [D18F3xE8\[HtcCapable\]=1](#)
- [D18F3x64\[HtcEn\]=1](#)
- PWROK=1
- THERMTRIP\_L=1

and any of the following conditions are true:

- Tctl is greater than or equal to the HTC temperature limit ([D18F3x64\[HtcTmpLmt\]](#)).
- PROCHOT\_L=0

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=1.

The default value of the HTC temperature threshold (Tctl\_max) is specified in the Power and Thermal Data-sheet.

### 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. See 2.5.3.1 [Core P-states].

The LHTC-active state is independent from the HTC-active state. LHTC does not affect PROCHOT\_L output and is not affected by PROCHOT\_L input.

The processor enters the LHTC-active state if all of the following conditions are true:

- PWROK is asserted.
- The processor is not in the package C6 (PC6) state.
- Tctl is greater than or equal to LHTC temperature limit.

The processor exits the LHTC-active state when the following is true:

- Tctl is less than LHTC temperature low limit since being greater than or equal to the LHTC temperature high limit.

### 2.10.3.3 Software P-state Limit Control

D18F3x68 [Software P-state Limit] provides a software mechanism to limit the P-state MSRC001\_0061[CurP-stateLimit]. See 2.5.3 [CPU Power Management].

### 2.10.3.4 THERMTRIP

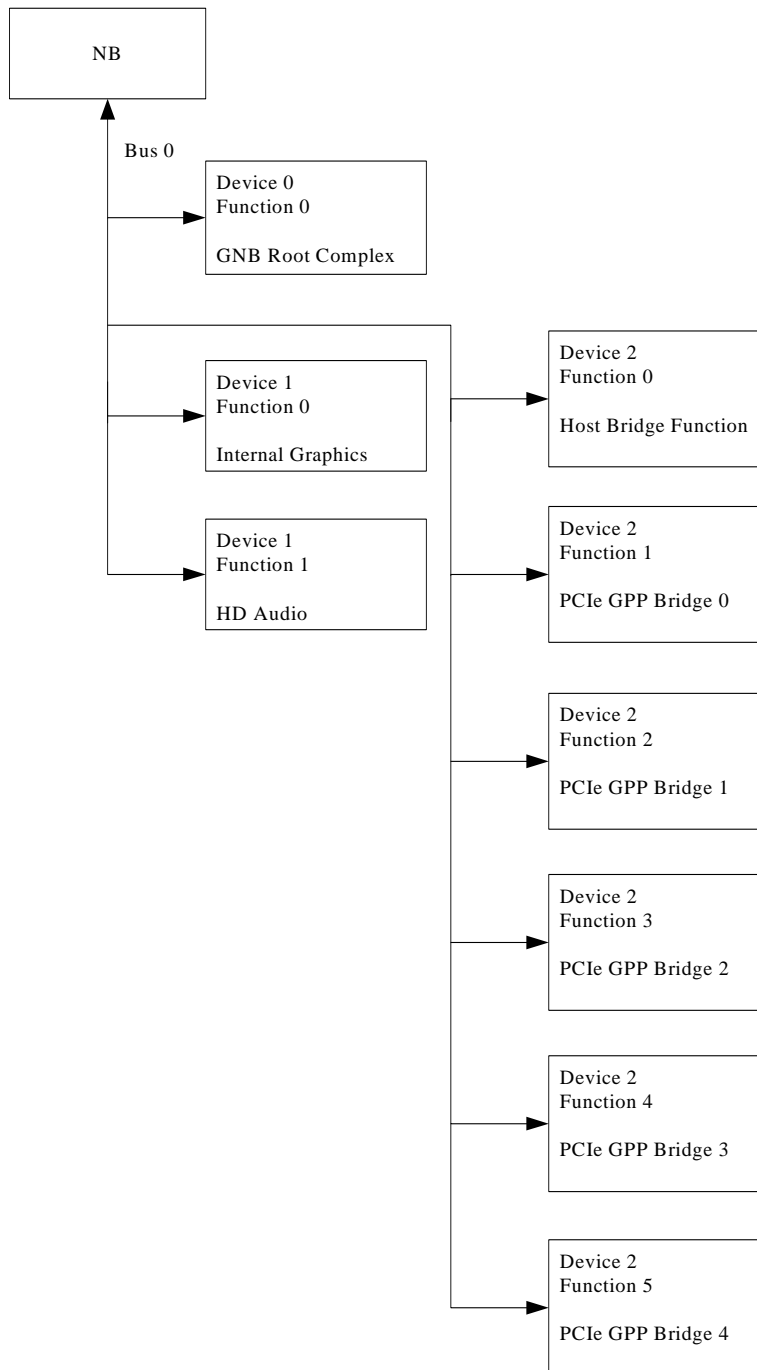
If the processor supports the THERMTRIP state (as specified by D18F3xE4[ThermtmpEn] or CPUID Fn8000\_0007\_EDX[TTP], which are the same) and the temperature approaches the point at which the processor may be damaged, the processor enters the THERMTRIP state. The THERMTRIP function is enabled after cold reset (after PWROK asserts and RESET\_L deasserts). It remains enabled in all other processor states, except during warm reset (while RESET\_L is asserted). The THERMTRIP state is characterized as follows:

- The THERMTRIP\_L signal is asserted.
- Nearly all clocks are gated off to reduce dynamic power.
- A low-value VID is generated.
- The system is placed into the S5 ACPI state (power off).

A cold reset is required to exit the THERMTRIP state.

## 2.11 Root Complex

### 2.11.1 Overview



**Figure 8: Root complex topology**

### 2.11.2 Interrupt Routing

The GNB includes a fully programmable IOAPIC. The IOAPIC registers are accessed through the [D0F0xF8](#) index and [D0F0xFC](#) data pair registers using two back-to-back config cycles. PCI defined INTx interrupts for



each bridge are routed to IOAPIC pins via the bridge interrupt routing registers located at [D0F0xFC\\_x1\[4:0\]](#).

### 2.11.2.1 IOAPIC Configuration

The IOAPIC configuration is performed by the following sequence:

1. Set the base address for the memory mapped registers by programming [D0F0xFC\\_x01\[IoapicAddr\]](#) and [D0F0xFC\\_x02\[IoapicAddrUpper\]](#).
2. Enable IOAPIC by programming [D0F0xFC\\_x00\[IoapicEnable\]](#) = 1.
3. Only if the system is in PIC mode, program [D0F0xFC\\_x00\[IoapicSbFeatureEn\]](#) = 1. This bit should be programmed to 0 when the system is in APIC mode.

The IOAPIC has a total of 29 interrupt inputs. These inputs are as follows:

- 5 groups of PCIe<sup>®</sup> interrupts each having a 4-bit external interrupt bus (INT A/B/C/D) and a 1-bit bridge interrupt, and
- a 4-bit external interrupt bus from GBIF.

The recommended interrupt routing and swizzling configuration is as shown in [Table 61](#).

**Table 61: Recommended Interrupt Routing and Swizzling Configuration**

Device	Register	Setting	Description
Dev2Fn1	<a href="#">D0F0xFC_x10[BrExtIntrGrp]</a>	0h	Map INT A/B/C/D to interrupt 0/1/2/3. Map bridge interrupt to interrupt 24.
	<a href="#">D0F0xFC_x10[BrExtIntrSwz]</a>	0h	
	<a href="#">D0F0xFC_x10[BrIntIntrMap]</a>	18h	
Dev2Fn2	<a href="#">D0F0xFC_x11[BrExtIntrGrp]</a>	1h	Map INT A/B/C/D to interrupt 4/5/6/7. Map bridge interrupt to interrupt 25.
	<a href="#">D0F0xFC_x11[BrExtIntrSwz]</a>	0h	
	<a href="#">D0F0xFC_x11[BrIntIntrMap]</a>	19h	
Dev2Fn3	<a href="#">D0F0xFC_x12[BrExtIntrGrp]</a>	2h	Map INT A/B/C/D to interrupt 8/9/10/11. Map bridge interrupt to interrupt 26.
	<a href="#">D0F0xFC_x12[BrExtIntrSwz]</a>	0h	
	<a href="#">D0F0xFC_x12[BrIntIntrMap]</a>	1Ah	
Dev2Fn4	<a href="#">D0F0xFC_x13[BrExtIntrGrp]</a>	3h	Map INT A/B/C/D to interrupt 12/13/14/15. Map bridge interrupt to interrupt 27.
	<a href="#">D0F0xFC_x13[BrExtIntrSwz]</a>	0h	
	<a href="#">D0F0xFC_x13[BrIntIntrMap]</a>	1Bh	
Dev2Fn5	<a href="#">D0F0xFC_x14[BrExtIntrGrp]</a>	4h	Map INT A/B/C/D to interrupt 16/17/18/19. Map bridge interrupt to interrupt 24.
	<a href="#">D0F0xFC_x14[BrExtIntrSwz]</a>	0h	
	<a href="#">D0F0xFC_x14[BrIntIntrMap]</a>	18h	
GBIF	<a href="#">D0F0xFC_x0F[GBIFExtIntrGrp]</a>	5h	Map INT A/B/C/D to interrupt 20/21/22/23.
	<a href="#">D0F0xFC_x0F[GBIFExtIntrSwz]</a>	0h	

### 2.11.3 Links

#### 2.11.3.1 Overview

There is one 5 x8 PCIe<sup>®</sup> core with five configurable ports. These ports can be divided into 2 groups:

- Gfx: Contains 1 x4 port to create a 4-bit Gfx lane using the upper 4 lanes of the PCIe<sup>®</sup> core.
- GPP: Contains upto 4 General Purpose Ports (GPP) using the lower 4 lanes of the PCIe<sup>®</sup> core.

All PCIe<sup>®</sup> links are capable of supporting Gen1/Gen2 data rates.

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 8](#).

Each PCIe® lane is assigned a unique lane ID that software uses to communicate configuration information to the SMU.

### 2.11.3.2 Link Configurations

Lanes of the Gfx ports can be assigned to IO links.

- For each 1x lane, program [D0F0xE4\\_x0130\\_0\[C:8\]04\[StrapBifMaxPayloadSupport\]=1h](#). The following link configurations are supported for the GPP links:

**Table 62: Supported General Purpose (GPP) Link Configurations**

D0F0xE4		GPP Port Lane				
x0130_0080	x0110_0011	Lanes[7:4]	3	2	1	0
0000_0001h	0000_0300h	x4 Link (Gfx)	x4 Link			
0000_0002h	0000_0203h	x4 Link (Gfx)	x2 Link		x2 Link	
0000_0003h	0000_0201h	x4 Link (Gfx)	x1 Link	x1 Link	x2 Link	
0000_0004h	0000_0200h	x4 Link (Gfx)	x1 Link	x1 Link	x1 Link	x1 Link

### 2.11.4 Root Complex Configuration

#### 2.11.4.1 LPC MMIO Requirements

To ensure proper operation of LPC generated DMA requests, the FCH must be configured to send processor generated MMIO writes that target the LPC bus to the FCH as non-posted writes. The MMIO address space of the LPC bus must not be included in the ranges specified by [D18F1x\[2CC:2A0,1CC:180,BC:80\] \[MMIO Base/Limit\]](#) and non-posted protocol for memory writes must be enabled using the following programming before LPC DMA transactions are initiated.

1. Program [D0F0x98\\_x06\[UmiNpMemWrEn\] = 1](#).

#### 2.11.4.2 Configuration for non-FCH Bridges

BIOS should program the following in non-FCH bridges:

1. Program [D0F0xCC\\_x01\[CrSEnable\] = 1](#) for [D0F0xC8\[NbDevIndSel\] = 11h-15h](#).
2. Program [D0F0xCC\\_x01\[SetPowEn\] = 1](#) for [D0F0xC8\[NbDevIndSel\] = 11h-15h](#).

#### 2.11.4.3 Link Configuration and Initialization

Link configuration and initialization is performed by the following sequence:

1. [2.11.4.3.1 \[Link Configuration and Core Initialization\]](#)
2. [2.11.4.3.2 \[Link Training\]](#)
3. [2.11.4.5 \[Link Power Management\]](#)
4. Lock link configuration registers.
  - Program [D0F0xE4\\_x0140\\_0010\[HwInitWrLock\] = 1](#).
  - Program [D0F0x64\\_x00\[HwInitWrLock\] = 1](#).

### 2.11.4.3.1 Link Configuration and Core Initialization

Link configuration is done on a per link basis. Lane reversal, IO link selection, and lane enablement is configured through this sequence.

1. Place software-reset module into blocking mode:
  - A. Program D0F0xE4\_x0130\_8062[ConfigXferMode]=0.
  - B. Program D0F0xE4\_x0130\_8062[BlockOnIdle]=0.
2. If the link is an IO link, Program D0F0xE4\_x0140\_0011[DynClkLatency]=Fh.
3. Program D0F0xE4\_x0130\_0080 per Table 62.
4. Program D0F0xE4\_x0130\_802[4:1] per Table 62.
5. Program D0F0xE4\_x0110\_0010[RxDetectTxPwrMode]=1.
6. Program D0F0xE4\_x0110\_0010[Ln2ExitTime]=000b.
7. Program D0F0xE4\_x0130\_8013[MasterPciePllA, MasterPciePllB] per Table 62.
8. Initiate core reconfiguration sequence:
  - A. Program D0F0xE4\_x0130\_8062[ReconfigureEn]=1.
  - B. Program D0F0xE4\_x0130\_8060[Reconfigure]=1.
  - C. Wait for D0F0xE4\_x0130\_8060[Reconfigure]==0.
  - D. Program D0F0xE4\_x0130\_8062[ReconfigureEn]=0.
9. Return software-reset module to non-blocking mode:
  - A. Program D0F0xE4\_x0130\_8062[ConfigXferMode]=1.
10. Program D2F[5:1]xE4\_xC1[StrapReverseLanes] if necessary.
11. Program D0F0xE4\_x0110\_0011 per Table 62.
12. For each nibble that has no PCIe<sup>®</sup> lanes in use:
  - A. Program D0F0xE4\_x0110\_001[8:7,3:2][PllPowerStateInOff]=111b.
  - B. Program D0F0xE4\_x0110\_001[8:7,3:2][PllPowerStateInTxs2]=111b.
  - C. Program D0F0xE4\_x0110\_001[8:7,3:2][TxPowerStateInTxs2]=111b.
  - D. Program D0F0xE4\_x0110\_001[8:7,3:2][RxPowerStateInRxs2]=111b.
13. For each lane that is not in use, program the corresponding D0F0xE4\_x0130\_8029[LaneEnable]=0.
14. Configure PIF pairings:
  - A. Program D0F0xE4\_x0110\_0011=0000\_0300h.

### 2.11.4.3.2 Link Training

Link training is performed on a per link basis. BIOS may train the links in parallel.

### 2.11.4.4 Miscellaneous Features

#### 2.11.4.4.1 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 D2F[5:1]xE4\_xC1[StrapReverseLanes]=1.
- To reverse the physical lane numbering for all ports in the GPP or GFX interfaces, program D0F0xE4\_x0140\_00C0[StrapReverseAll]=1.

Note that logical port numbering is established during link training regardless of the physical lane numbering.

#### 2.11.4.4.2 Link Speed Changes

Link speed changes can only occur on Gen2 capable links. To verify that Gen2 speeds are supported verify D2F[5:1]x64[LinkSpeed]==02h.

#### 2.11.4.4.2.1 Autonomous Link Speed Changes

To enable autonomous speed changes on a per port basis:

1. Program `D2F[5:1]xE4_x88[TargetLinkSpeed]=2h`.
2. Program `D0F0xE4_x0130_0[C:8]03[StrapBifDeemphasisSel]=1`.
3. Program `D2F[5:1]xE4_xA4[LcGen2EnStrap]=1`.
4. Program `D2F[5:1]xE4_xC0[StrapAutoRcSpeedNegotiationDis]=0`.
5. Program `D2F[5:1]xE4_xA4[LcMultUpstreamAutoSpdChngEn]=1`.
6. Program `D2F[5:1]xE4_xA2[LcUpconfigureDis]=0`.

#### 2.11.4.4.3 Deemphasis

Deemphasis strength can be changed on a per-port basis by programming `D2F[5:1]xE4_xB5[LcSelectDeemphasis]`.

#### 2.11.4.5 Link Power Management

##### 2.11.4.5.1 Link States

To enable support for L1 program `D2F[5:1]xE4_xA0[LcL1Inactivity]=6h`.

To enable support for L0s:

- Program `D2F[5:1]xE4_xA1[LcDontGotoL0sifL1Armed]=1`.
- Program `D2F[5:1]xE4_xA0[LcL0sInactivity]=9h`.

##### 2.11.4.5.2 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 link can switch among widths of: x1, x2, and x4, upto the maximum programmed port width. The GPP link can switch among widths of: x1, x2, and x4, upto the maximum programmed port width. The link width is controlled by the following mechanism:

- 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 `D2F[5:1]xE4_xA2[LcDynLanesPwrState]=11b`.

#### 2.11.4.6 Link Test and Debug Features

##### 2.11.4.6.1 Compliance Mode

To enable Gen1 software compliance mode program `D2F[5:1]xE4_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_x0140_00C1[StrapGen2Compliance]=1`.
2. Program `D2F[5:1]xE4_xA4[LcGen2EnStrap]=1`.
3. Program `D2F[5:1]xE4_x88[TargetLinkSpeed]=2h` for each port to be placed in compliance mode.

4. Program `D2F[5:1]x88[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.

### 2.11.6 PCIe® Client Interface Control

This interface is accessed through the indexed space registers located at `D0F2xF8` within the [Device 0 Function 2 Configuration Registers](#).

BIOS should perform the following steps to initialize the interface:

1. Program `D0F0x64_x0D[PciDev0Fn2RegEn] = 1h`.
2. Program credits for the BIF client as follows:
  - A. Program `D0F2xFC_x32_L1i[1][DmaNpHaltDis] = 1h`.
  - B. Program `D0F2xFC_x32_L1i[1][DmaBufCredits] = 8h`.
  - C. Program `D0F2xFC_x32_L1i[1][DmaBufMaxNpCred] = 8h`.
3. Program credits for the PPD client as follows:
  - A. Program `D0F2xFC_x32_L1i[0][DmaBufCredits] = 8h`.
  - B. Program `D0F2xFC_x32_L1i[0][DmaBufMaxNpCred] = 7h`.
4. Program credits for the INTGEN client as follows:
  - A. Program `D0F2xFC_x32_L1i[2][DmaBufCredits] = 4h`.
  - B. Program `D0F2xFC_x32_L1i[2][DmaBufMaxNpCred] = 4h`.
5. Program clock gating as follows:
  - A. Program `D0F2xFC_x33[L1DmaClkgateEn] = 1h`.
  - B. Program `D0F2xFC_x33[L1CacheClkgateEn] = 1h`.
  - C. Program `D0F2xFC_x33[L1CpslvClkgateEn] = 1h`.
  - D. Program `D0F2xFC_x33[L1DmaInputClkgateEn] = 1h`.
  - E. Program `D0F2xFC_x33[L1PerfClkgateEn] = 1h`.
  - F. Program `D0F2xFC_x33[L1MemoryClkgateEn] = 1h`.
  - G. Program `D0F2xFC_x33[L1RegClkgateEn] = 1h`.
  - H. Program `D0F2xFC_x33[L1HostreqClkgateEn] = 1h`.
  - I. Program `D0F2xFC_x33[L1L2ClkgateEn] = 1h`.
6. Program `D0F0x64_x0D[PciDev0Fn2RegEn] = 0h`.

## 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. The SMU contains a microcontroller to assist with many of these tasks.

### 2.12.1 Software Interrupts

The microcontroller can be interrupted to cause it to perform several initialization and runtime tasks. BIOS and ACPI methods can interrupt the SMU to request a specific action using the following sequence:

1. If a service request requires an argument, program `D0F0xBC_xC210_003C[Argument]` with the desired argument.
1. Wait for `D0F0xBC_xC210_0004[IntDone]==1`.
2. Program `D0F0xBC_xC210_0000[ServiceIndex]` to the desired service index and toggle. This may be done in a single write.
3. Wait for `D0F0xBC_xC210_0004[IntAck]==1`.

After performing the steps above, software may continue execution before the interrupt has been serviced. However, software should not rely on the results of the interrupt until the service is complete (see `D0F0xBC_xC210_0004[IntDone]`). Interrupting the SMU with a service index that does not exist results in undefined behavior.

**Table 63: SMU Software Interrupts**

Service Index	Notes
1Eh	Description: <b>BIOSSMC_MSG_LCLK_DPM_ENABLE</b> . Enables LCLK DPM. See 2.5.6.1.3 [LCLK DPM].
	Input: <code>D0F0xBC_x3FDC8</code>
	Output: None.
3Ah	Description: <b>BIOSSMC_MSG_VDDNB_REQUEST</b> . Request VDDNB voltage.
	Input: Program argument to desired voltage (encoded in mV with two fraction bits). Values outside <code>D18F5x17C[MaxVid, MinVid]</code> range are invalid and result in undefined behavior. <ul style="list-style-type: none"> <li>• <code>D0F0xBC_xC210_003C[Argument] = (Desired Voltage in mV) * 4</code>.</li> </ul>
	Output: None.
43h	Description: <b>BIOSSMC_MSG_NBDPM_Enable</b> . Enables NB P-state Adjustments.
	Input: <code>D0F0xBC_x3F9E8</code>
	Output: <code>D0F0xBC_x3F9EC</code>

## 2.13 Graphics Processor (GPU)

The APU contains an integrated DX11 compliant graphics processor.

### 2.13.1 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.

### 2.13.2 Frame Buffer (FB)

The frame buffer is defined as the portion of system memory dedicated for GPU use.

**Table 64: Recommended Frame Buffer Configurations**

System Memory Size	Frame Buffer Size
< 1GB	64 MB
>= 1GB & < 2GB	256 MB
>= 2GB	512 MB

## 2.14 RAS Features

### 2.14.1 Machine Check Architecture

The processor contains logic and registers to detect, log, and correct errors in the data or control paths in each core and the Northbridge. The Machine Check Architecture (MCA) defines the facilities by which processor and system hardware errors are logged and reported to system software. This allows system software to perform a strategic role in recovery from and diagnosis of hardware errors.

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].

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\[MCE\]](#).

#### 2.14.1.1 Machine Check Registers

[CPUID Fn0000\\_0001\\_EDX\[MCA\]](#) or [CPUID Fn8000\\_0001\\_EDX\[MCA\]](#) indicates the presence of the following machine check registers:

- [MSR0000\\_0179 \[Global Machine Check Capabilities \(MCG\\_CAP\)\]](#)
  - Reports how many machine check register banks are supported.
- [MSR0000\\_017A \[Global Machine Check Status \(MCG\\_STAT\)\]](#)
  - Provides basic information about processor state after the occurrence of a machine check error.
- [MSR0000\\_017B \[Global Machine Check Exception Reporting Control \(MCG\\_CTL\)\]](#)
  - Used by software to enable or disable the logging and reporting of machine check errors in the error-reporting banks.

The error-reporting machine check register banks supported in this processor are:

- MC0: Data cache (DC).
  - MC1: Instruction cache (IC).
  - MC2: Bus unit (BU), including L2 cache.
  - MC3: Reserved.
  - MC4: Northbridge (NB), including the IO link. These MSRs are also accessible from configuration space. There is only one NB error-reporting bank, independent of the number of cores.
  - MC5: Fixed-issue reorder buffer (FR) machine check registers.
- The register types within each bank are:
    - [MCi\\_CTL](#), Machine Check Control: Enables error reporting via machine check exception. The [MCi\\_CTL](#) register in each bank must be enabled by the corresponding enable bit in [MCG\\_CTL \(MSR0000\\_017B\)](#).
    - [MCi\\_STATUS](#), Machine Check Status: Logs information associated with errors.
    - [MCi\\_ADDR](#), Machine Check Address: Logs address information associated with errors.
    - [MCi\\_MISC](#), Machine Check Miscellaneous: Log miscellaneous information associated with errors, as defined by each error type.
    - [MCi\\_CTL\\_MASK](#), Machine Check Control Mask: Inhibit detection of an error source unless otherwise specified.

The following table identifies the registers associated with each error-reporting machine check register bank:



**Table 65: MCA register cross-reference table**

Register Bank (MC <sub>i</sub> )	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

Corrected, deferred, and uncorrected errors are logged in MC<sub>i</sub>\_STATUS and MC<sub>i</sub>\_ADDR as they occur. Uncorrected errors that are enabled in MC<sub>i</sub>\_CTL result in a Machine Check exception.

Each MC<sub>i</sub>\_CTL register must be enabled by the corresponding enable bit in MSR0000\_017B [Global Machine Check Exception Reporting Control (MCG\_CTL)].

MC<sub>i</sub>\_CTL\_MASK allow BIOS to mask the presence of any error source 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 and actions.

Each MCA bank implements a number of machine check miscellaneous registers, denoted as MC<sub>i</sub>\_MISC<sub>j</sub>, where j goes from 0 to a maximum of 8. If there is more than one MC<sub>i</sub>\_MISC register in a given bank, a non-zero value in MC<sub>i</sub>\_MISC0[BlkPtr] points to the contiguous block of additional registers.

The presence of valid information in the first MISC register in the bank (MC<sub>i</sub>\_MISC0) is indicated by MC<sub>i</sub>\_STATUS[MiscV]. The presence of valid information in additional implemented MISC registers is indicated by MC<sub>i</sub>\_MISC<sub>j</sub>[Val] in the target register.

#### 2.14.1.2 Machine Check Errors

The classes of machine check errors are, in priority order from highest to lowest:

- Uncorrected
- Deferred
- Corrected

Uncorrected errors cannot be corrected by hardware and may cause loss of data, corruption of processor state, or both. Uncorrected errors update the status and address registers if not masked from logging in MC<sub>i</sub>\_CTL\_MASK. Information in the status and address registers from a previously logged lower priority error is overwritten. Previously logged errors of the same priority are not overwritten. Uncorrected errors that are enabled in MC<sub>i</sub>\_CTL result in reporting to software via machine check exceptions. If an uncorrected error is masked from logging, the error is ignored by hardware (exceptions are noted in the register definitions). If an uncorrected error is disabled from reporting, containment of the error and logging/reporting of subsequent errors may be affected. Therefore, enable reporting of unmasked uncorrected errors for normal operation. Disable reporting of uncorrected errors only for debug purposes.

Deferred errors are errors that cannot be corrected by hardware, but do not cause an immediate interruption in

program flow, loss of data integrity, or corruption of processor state. These errors indicate that data has been corrupted but not consumed; no exception is generated because the data has not been referenced by a core or an IO link. Hardware writes information to the status and address registers in the corresponding bank that identifies the source of the error if deferred errors are enabled for logging. If there is information in the status and address registers from a previously logged lower priority error, it is overwritten. Previously logged errors of the same or higher priority are not overwritten. Deferred errors are not reported via machine check exceptions; they can be seen by polling the MCI\_STATUS registers.

Corrected errors are those which have been corrected by hardware and cause no loss of data or corruption of processor state. Hardware writes the status and address registers in the corresponding register bank with information that identifies the source of the error if they are enabled for logging. Corrected errors are not reported via machine check exceptions. Some corrected errors may be reported to software via error thresholding (see 2.14.1.7 [Error Thresholding]).

The implications of these categories of errors are:

1. Uncorrected error; hardware did not deal with the problem.
  - Operationally (error handling), action required, because program flow is affected.
  - Diagnostically (fault management), software may collect information to determine if any components should be de-configured or serviced.
2. Deferred error; hardware partially dealt with the problem via containment.
  - Operationally, action optional, because program flow has not been affected. However, steps may be taken by software to prevent access to the data in error.
  - Diagnostically, software may collect information to determine if any components should be de-configured or serviced.
3. Corrected error; hardware dealt with the problem.
  - Operationally, no action required, because program flow is unaffected.
  - Diagnostically, software may collect information to determine if any components should be de-configured or serviced.

Machine check conditions can be simulated to aid in debugging machine check handlers. See 2.14.3 [Error Injection and Simulation] for more detail.

### 2.14.1.3 Error Detection, Action, Logging, and Reporting

Error detection is controlled by the MASK registers:

- Error detection for MCA controlled errors is enabled if not masked by MCI\_CTL\_MASK (see Table 65 [MCA register cross-reference table]).
- Error masking is performed regardless of MCA bank enablement in MCG\_CTL (MSR0000\_017B).

Error action refers to the hardware response to an error, aside from logging and reporting. Enablement of error action for each error is enumerated in the EAC (Error Action Condition) column of the error descriptions tables as follows:

- D: Detected. The error action is taken if the error is detected (i.e., not masked). These actions occur regardless of whether the MCA bank is enabled in MCG\_CTL.
- E: Enabled. The error action is taken if the error is detected and the bank is enabled in MCG\_CTL.

Error logging refers to the storing of information in the status registers, and is enabled if all of the following are true:

- Error detection is enabled.
- The MCA bank is enabled in MCG\_CTL.

Error reporting refers to active notification of errors to software via machine check exceptions, and is enabled if all of the following are true:

- Error logging is enabled.
- The corresponding enable bit for the error in `MCi_CTL` is set to 1.

A machine check exception will be generated if all the following are true:

- The error is uncorrected.
- The error is enabled for reporting.
- `CR4.MCE` is enabled.

Notes:

1. If `CR4.MCE` is clear, an error configured to cause a machine check exception will cause a shutdown.
2. If error reporting is disabled, the setting of `CR4.MCE` has no effect.
3. If an uncorrected error is disabled from reporting, containment of the error and logging/reporting of subsequent errors may be affected. Therefore, unmasked uncorrected errors should be enabled for reporting for normal operation. Uncorrected errors should only be disabled from reporting for debug purposes.
4. Errors not associated with a specific core are reflected to core 0 of the compute unit. The error description tables identify which errors are associated or not associated with a specific core of the compute unit.

Throughout the MCA register descriptions, the terms “enabled” and “disabled” generally refer to reporting, and the terms “masked” and “unmasked” generally refer to logging, unless otherwise noted.

Some logged errors increment a counter in `MCi_MISC`, which may trigger an interrupt (see [2.14.1.7 \[Error Thresholding\]](#)). Although no machine check exception will be generated, these notifications can be viewed as “correctable machine check interrupts”.

For debug observability only, `D18F3x180[ConvertUnCorToCorErrEn]` can be used to log NB uncorrected errors as corrected errors.

#### 2.14.1.3.1 MCA conditions that cause Shutdown

The following architectural conditions cause the processor to enter the Shutdown state; see section “Machine-Check Errors” in APM volume 2 for more detail; see [1.2 \[Reference Documents\]](#):

- Attempting to generate an MCE when machine check reporting is disabled at the system level (`CR4.MCE=0`).
- Attempting to generate an MCE when a machine check is in progress on the same core (`MSR0000_017A[MCIP]=1`).

The following non-architectural conditions cause the processor to enter the Shutdown state:

- On MCA interrupts, if two or more uncorrected errors occur across MCA banks.

#### 2.14.1.3.2 Error Logging During Overflow

An error to be logged when the status register contains valid data can result in an overflow condition. During error overflow conditions, the new error may not be logged or an error which has already been logged in the status register may be overwritten. For the rules on error overflow, priority, and overwriting, see [MSR0000\\_0401\[Overflow\]](#).

Overflow alone does not indicate a shutdown condition. Uncorrected errors require software intervention. Therefore, when an uncorrected error cannot be logged, critical error information may have been lost, and `MCi_STATUS[PCC]` may be set. If PCC is indicated, software should terminate system processing to prevent

data corruption (see 2.14.1.6 [Handling Machine Check Exceptions]). If PCC is not indicated, any MCA data lost due to overflow was informational only and not critical to system hardware operation.

**Table 66: Overwrite Priorities for All Banks**

			Older Error			
			Uncorrected		Corrected	
			Enabled	Disabled	Enabled	Disabled
Newer Error	Uncorrected	Enabled	-	-	Overwrite	Overwrite
		Disabled	-	-	Overwrite	Overwrite
	Corrected	Enabled	-	-	-	-
		Disabled	-	-	-	-

**2.14.1.4 MCA Initialization**

The following initialization sequence must be followed:

- MCI\_CTL\_MASK registers (see Table 65 [MCA register cross-reference table] for list):
  - BIOS must initialize the mask registers to inhibit error detection prior to the initialization of MCI\_CTL and MSR0000\_017B.
  - BIOS must not clear MASK bits that are reset to 1.
- The MCI\_CTL registers must be initialized by the operating system prior to enabling the error reporting banks in MCG\_CTL.

If initializing after a cold reset (see D18F0x6C[ColdRstDet]), then BIOS must clear the MCI\_STATUS MSRs. If initializing after a warm reset, then BIOS should check for valid MCA errors and if present save the status for later diagnostic use (see 2.14.1.6 [Handling Machine Check Exceptions]).

BIOS may initialize the MCA without setting CR4.MCE; this will result in a system shutdown on any machine check which would have caused a machine check exception (followed by a reboot if configured in the chipset). Alternatively, BIOS that wishes to ensure continued operation in the event that a machine check occurs during boot may write MCG\_CTL with all ones and write zeros into each MCI\_CTL. With these settings, a machine check error will result in MCI\_STATUS being written without generating a machine check exception or a system shutdown. BIOS may then poll MCI\_STATUS during critical sections of boot to ensure system integrity. Before passing control to the operating system, BIOS should restore the values of those registers to what the operating system is expecting. (Note that using MCI\_CTL to disable error reporting on uncorrected errors may affect error containment; see 2.14.1.3 [Error Detection, Action, Logging, and Reporting].)

Before ECC memory has been initialized with valid ECC check bits, BIOS must ensure that no memory operations are initiated if MCA reporting is enabled. This includes memory operations that may be initiated by hardware prefetching or other speculative execution. It is recommended that, until all of memory has been initialized with valid ECC check bits, the BIOS either does not have any valid MTRRs specifying a DRAM memory type or does not enable DRAM ECC machine check exceptions.

**2.14.1.5 Error Code**

The MCI\_STATUS[ErrorCode] field contains information used to identify the logged error. Table 67 [Error Code Types] identifies how to decode ErrorCode. The MCI\_STATUS[ErrorCodeExt] field contains detailed, model-specific information that is used to further narrow identification for error diagnosis, but not error handling by software; see 2.14.1.6 [Handling Machine Check Exceptions].

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; the value of ErrorCodeExt is unique within Error Code Type. Details for each Error Type are described in the tables accompanying the MCi\_STATUS register for each bank.

- MC0 (DC); Table 197 [MC0 Error Signatures].
- MC1 (IC); Table 200 [MC1 Error Signatures].
- MC2 (BU); Table 203 [MC2 Error Signatures].
- MC4 (NB); Table 206 [MC4 Error Signatures, Part 1] and Table 207 [MC4 Error Signatures, Part 2].
- MC5 (FR); Table 215 [MC5 Error Signatures].

**Table 67: Error Code Types**

Error Code	Error Code Type	Description
0000 0000 0001 TTLL	TLB	TT = Transaction Type LL = Cache Level
0000 0001 RRRR TTLL	Memory	Errors in the cache hierarchy (not in NB) RRRR = Memory Transaction Type TT = Transaction Type LL = Cache Level
0000 1PPT RRRR IILL	Bus	General bus errors including link and DRAM PP = Participation Processor T = Timeout RRRR = Memory Transaction Type II = Memory or IO LL = Cache Level
0000 01UU 0000 0000	Internal Unclassified	Internal unclassified errors UU = Internal Error Type

**Table 68: Error codes: transaction type (TT)**

TT	Transaction Type
00	Instr: Instruction
01	Data
10	Gen: Generic
11	Reserved

**Table 69: Error codes: cache level (LL)**

LL	Cache Level
00	Reserved
01	L1: Level 1
10	L2: Level 2
11	LG: Generic

**Table 70: Error codes: memory transaction type (RRRR)**

RRRR	Memory Transaction Type
0000	Gen: Generic. Includes scrub errors.

**Table 70: Error codes: memory transaction type (RRRR)**

RRRR	Memory Transaction Type
0001	RD: Generic Read
0010	WR: Generic Write
0011	DRD: Data Read
0100	DWR: Data Write
0101	IRD: Instruction Fetch
0110	Prefetch
0111	Evict
1000	Snoop (Probe)

**Table 71: Error codes: participation processor (PP)**

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 72: Error codes: memory or IO (II)**

II	Memory or IO
00	Mem: Memory Access
01	Reserved
10	IO: IO Access
11	Gen: Generic

**Table 73: Error codes: Internal Error Type (UU)**

UU	Internal Error Type
00	Reserved
01	Reserved
10	HWA: Hardware Assertion
11	Reserved

#### 2.14.1.6 Handling Machine Check Exceptions

A machine check handler is invoked to handle an exception for a particular core. Because MCA registers are generally not shared among cores, the handler does not need to coordinate register usage with handler instances on other cores. Those few MCA registers which are shared are noted in the register description. (See also [2.4.2.1 \[Registers Shared by Cores in a L2 complex\].](#))

For access to the core MC2 registers, [MSRC001\\_10A0\[McaToMstCoreEn\]](#) allows a single core to access the registers through MSR space without contention from other cores. This organization of registers on a per core basis allows independent execution, simplifies exception handling, and reduces the number of conditions which are globally fatal.

For access to the NB MCA registers, [D18F3x44\[NbMcaToMstCpuEn\]](#) allows a single core (the NBC) to access the registers through MSR space without contention from other cores. This organization of registers on a per core basis allows independent execution, simplifies exception handling, and reduces the number of conditions which are globally fatal.

At a minimum, the machine check handler must be capable of logging error information 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 by software. If a recoverable error is identified, the exception handler can attempt to correct the error and restart the interrupted program. An error may not be recoverable for the process or virtual machine it directly affects, but may be containable, so that other processes or virtual machines in the system are unaffected and system operation is recovered; see [2.14.1.6.1 \[Differentiation Between System-Fatal and Process-Fatal Errors\]](#).

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:

- Data collection:
  - All status registers in all error reporting banks must be examined to identify the cause of the machine check exception.
    - Read [MSR0000\\_0179\[Count\]](#) to determine the number of status registers visible to the 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. These are generically referred to as MC<sub>i</sub>\_STATUS.
    - Check the valid bit in each status register (MC<sub>i</sub>\_STATUS[Val]). The remainder of the status register should be examined only when its valid bit is set.
    - When identifying the error condition and determining how to handle the error, portable exception handlers should examine the following MC<sub>i</sub>\_STATUS fields: ErrorCode, UC, PCC, CECC, UECC, Deferred, Poison. The expected settings of these and other fields in MC<sub>i</sub>\_STATUS are identified in the error signatures tables which accompany the descriptions of each MCA status register. See [2.14.1.5 \[Error Code\]](#) for a discussion of error codes and pointers to the error signatures tables.
      - MC<sub>i</sub>\_STATUS[ErrorCodeExt] should generally not be used by portable code to identify the error condition because it is model specific. ErrorCodeExt is useful in determining the error sub-type for root cause analysis.
    - Error handlers should collect all available MCA information (status register, address register, miscellaneous register, etc.), but should only interrogate details to the level which affects their actions. Lower level details may be useful for diagnosis and root cause analysis, but not for error handling.
- Recovery:
  - Check the valid MC<sub>i</sub>\_STATUS registers to see if error recovery is possible.
    - Error recovery is not possible when the processor context corrupt indicator (MC<sub>i</sub>\_STATUS[PCC]) is set to 1.
      - The error overflow status indicator (MC<sub>i</sub>\_STATUS[Overflow]) does not indicate whether error recovery is possible. See [2.14.1.3.2 \[Error Logging During Overflow\]](#).
    - If error recovery is not possible, the handler should log the error information and return to the operating system for system termination.
  - Check MC<sub>i</sub>\_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 error affects only process data, it may be possible to terminate only the affected process or virtual machine. If the error affects processor state, continued use of that processor should not occur. See individual error descriptions for further guidance.

- If [MSR0000\\_017A\[RIPV\]](#) is set, the interrupted program can be restarted reliably at the instruction pointer address pushed onto the exception handler stack if any uncorrected error has been corrected by software. If RIPV is clear, the interrupted program cannot be restarted reliably, although it may be possible to restart it for debugging purposes. As long as PCC is clear, it may be possible to terminate only the affected process or virtual machine.
- When logging errors, particularly those that are not recoverable, check [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 ensured to be related to the error.
- See [2.14.1.6.1 \[Differentiation Between System-Fatal and Process-Fatal Errors\]](#) for more explanation on the relationship between PCC, RIPV, and EIPV.
- Exit:
  - When an exception handler is able to successfully log an error condition, clear the `MCi_STATUS` registers prior to exiting the machine check handler. Software is responsible for clearing at least `MCi_STATUS[Val]`.
  - Prior to exiting the machine check handler, be sure to clear [MSR0000\\_017A\[MCIP\]](#). MCIP indicates that a machine check exception is in progress. If this bit is set when another machine check exception occurs in the same core, the processor enters the shutdown state.

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`.

In cases where sync flood is the recommended response to a particular error, a machine check exception cannot be used in lieu of the sync flood to stop the propagation of potentially bad data.

#### 2.14.1.6.1 Differentiation Between System-Fatal and Process-Fatal Errors

The bits `MCi_STATUS[PCC]`, [MSR0000\\_017A\[RIPV\]](#), and [MSR0000\\_017A\[EIPV\]](#) form a hierarchy, used by software to determine the degree of corruption and recoverability in the system.

[Table 74](#) shows how these bits are interpreted.

**Table 74: Error Scope Hierarchy**

PCC	UC	RIPV	EIPV	Deferred	Poison	Comments
1	1	-	-	-	-	<b>System fatal error.</b> Signaled via machine check exception, action required. Error has corrupted system state (PCC=1). The error is fatal to the system and the system processing must be terminated.
0	1	1	1	-	-	



**Table 74: Error Scope Hierarchy**

PCC	UC	RIPV	EIPV	Deferred	Poison	Comments
0	1	0	-	-	0/1	<b>Hardware uncorrected, software containable error.</b> Signaled via machine check exception, action required. The error is confined to the process, however the process cannot be restarted even if the uncorrected error is corrected by software.  Poison=1; the error is due to consumption of poison data. If the affected process or virtual machine is terminated, the system may continue operation.
0	0	-	-	1	0	<b>Deferred error.</b> Action optional. A latent error has been discovered, but not yet consumed; a machine check exception will be generated if the affected data is consumed. Error handling software may attempt to correct this data error, or prevent access by processes which map the data, or make the physical resource containing the data inaccessible. Note: May be detected on a demand access or a scrub access.
0	0	-	-	0	0	<b>Corrected error.</b> Signaled via error thresholding mechanisms (2.14.1.7 [Error Thresholding]); no action required.

### 2.14.1.7 Error Thresholding

For some types of errors, the hardware maintains counts of the number of errors. When the counter reaches a programmable threshold, an event may optionally be triggered to signal software. This is known as error thresholding. The primary purpose of error thresholding is to help software recognize an excessive rate of errors, which may indicate marginal or failing hardware. This information can be used to make decisions about deconfiguring hardware or scheduling service actions. Counts are incremented for corrected, deferred, and uncorrected errors.

The error thresholding hardware counts only the number of errors; it is up to software to track the errors reported over time in order to determine the rate of errors. Thresholding gives error counts on groups of resources. In order to make decisions on individual resources, a finer granularity of error information, such as MCA information for specific errors, must be utilized in order to obtain more accurate counts and to limit the scope of actions to affected hardware.

Thresholding is performed for “Error Threshold Groups” identified in the list below. For all error threshold groups, some number of corrected errors is expected and normal. There are numerous factors influencing error rates, including temperature, voltage, operating speed, and geographic location. In order to accommodate the various factors, including software latency to respond and track the error thresholding, additional guardband above the normal rates is recommended before error rates are considered abnormal for purposes of hardware action.

The {MC0, MC1, MC2, MC5} error thresholding banks maintains counters, but do not provide interrupts when the threshold is reached; these counters must be polled.

Error thresholding groups:

- DRAM (MC4)
  - Memory errors are counted and polled or reported via [MSR0000\\_0413](#).
  - DRAM errors are the errors listed in [Table 205 \[MC4 Error Descriptions\]](#) as “D” (DRAM) in the ETG (Error Threshold Group) column.
  - Operating systems can avoid or stop using memory pages with excessive errors.
- Links (MC4)
  - Link errors are counted and reported via [MSRC000\\_0408](#).
  - Link errors are the errors listed in [Table 205 \[MC4 Error Descriptions\]](#) as “L” (Cache) in the ETG (Error Threshold Group) column.
  - For a link exhibiting excessive errors, it may be possible to reduce errors by lowering the link frequency or reducing the link width (if a bad lane can be avoided). See [2.11 \[Root Complex\]](#) for details and restrictions on configuring links.

In rare circumstances, such as two simultaneous errors in the same error thresholding group, it is possible for one error not to increment the counter. In these conditions, `MCi_STATUS[Overflow]` may indicate that an overflow occurred, but the error counter may only indicate one error.

#### 2.14.1.8 Scrub Rate Considerations

This section gives guidelines for the scrub rate settings available in [D18F3x58 \[Scrub Rate Control\]](#) and [MSRC001\\_10A0 \[L2I Configuration \(L2I\\_CFG\)\]](#). Scrubbers are used to periodically read cacheline sized data locations and associated tags. There are two primary benefits to scrubbing. First, scrubbing corrects any corrected errors which are discovered before they can migrate into uncorrected errors. This is particularly important for soft errors, which are caused by external sources such as radiation and which are temporary conditions which do not indicate malfunctioning hardware. Second, scrubbers help identify marginal or failed hardware by finding and logging repeated errors at the same locations (see also [2.14.1.7 \[Error Thresholding\]](#)).

There are many factors which influence scrub rates. Among these are:

- The size of memory or cache to be scrubbed
- Resistance to upsets
- Geographic location and altitude
- Alpha particle contribution of packaging
- Performance sensitivity
- Risk aversion

The baseline recommendations in [Table 75](#) are intended to provide excellent protection at most geographic locations, while having no measurable effect on performance. Adjustments may be necessary due to special circumstances. Refer to JEDEC standards for guidelines on adjusting for geographic location.

**Table 75: Recommended Scrub Rates per Node**

Register	Memory Size per Node (GB)	Register Setting	Scrub Rate
<a href="#">MSRC001_10A0[L2ScrubberInterval]</a>	-	100h	98.6 ms

**Table 75: Recommended Scrub Rates per Node**

Register	Memory Size per Node (GB)	Register Setting	Scrub Rate
D18F3x58[DramScrub]	0 GB == Size	00h	Disabled
	0 GB < Size <= 1 GB	12h	5.24 ms
	1 GB < Size <= 2 GB	11h	2.62 ms
	2 GB < Size <= 4 GB	10h	1.31 ms
	4 GB < Size <= 8 GB	0Fh	655.4 us
	8 GB < Size <= 16 GB	0Eh	327.7 us
	16 GB < Size <= 32 GB	0Dh	163.8 us
	32 GB < Size <= 64 GB	0Ch	81.9 us
	64 GB < Size <= 128 GB	0Bh	41.0 us
	128 GB < Size <= 256 GB	0Ah	20.5 us
	256 GB < Size	09h	10.2 us

For steady state operation, finding a range of useful scrub rates may be performed by selecting a scrub rate which is high enough to give good confidence about protection from accumulating errors and low enough that it has no measurable effect on performance. The above baselines are made to maximize error coverage without affecting performance and not based on specific processor soft error rates.

For low power states in which the processor core is halted, the power management configuration may affect scrubbing; see 2.8.3 [Memory Scrubbers].

#### 2.14.1.9 Error Diagnosis

This section describes generalized information and algorithms for diagnosing errors. The primary goal of diagnosis is to identify the failing component for repair purposes. The secondary goal is to identify the smallest possible sub-component for deallocation, deconfiguration, or design/manufacturing root cause analysis.

*Indictment* means identifying the part in error. The simplest form of indictment is *self-indictment*, where the bank reporting the error is the faulty unit. The next simplest form of indictment is *eyewitness indictment*, where the faulty unit is not the bank reporting the error, but is identified unambiguously. Both of these forms can be considered direct indictment; the information for indictment is contained in the MCA error information. If an error is not directly indicted, then identifying the faulty unit is more difficult and may not be an explicit part of the error log.

In general, an address logged in the MCA is useful for direct indictment only if the address identifies a physical location in error, such as a cache index. Logical addresses, while identifying the data, do not identify the location of the data.

If possible, physical storage locations in caches should be checked to determine whether the error is a soft error (a temporary upset of the stored value) or a hard fault (malfunctioning hardware). A location which has had a soft error can be corrected by writing a new value to the location; a reread of the location should see the new value. Hard faults cannot be corrected by writing a new value; the hardware persistently returns the previous value. If such checking is not possible, a grossly simplifying assumption can be made that uncorrected errors are hard and corrected errors are soft. Repeated corrected errors from the same location are an indication that the fault is actually hard.

Determining whether corrected errors represent a hard fault or a soft error requires understanding the access

patterns and any attempts to correct the faulty data in place. An attempt to correct the data in place creates two *epochs*, one before the correction event, and one after. If an error is seen at the same location in two different epochs (especially back-to-back epochs), it is more likely that the cause is a hard fault, since the error has persisted or repeated through an in place correction. The more epochs in which a error is seen, the higher the likelihood of it being caused by a hard fault.

As an example, consider a corrected error found during a read from DRAM. If the DRAM redirect scrubber is enabled (`D18F3x5C[ScrubReDirEn]`), the data in error is corrected in place, and this event conceptually creates a new epoch. If the original fault was due to a soft error, a read of the same data in the new epoch should not encounter a data error. If the original fault was due to a hard fault (e.g., a stuck bit), a read of the data in the new epoch will likely result in another corrected or uncorrected error.

There are numerous correction events that can be used to separate time periods into epochs. These include DRAM redirect scrubs, DRAM sequential scrubs, cache scrubs, cache writes, cache flushes, resets, and others.

#### 2.14.1.9.1 Common Diagnosis Information

A common set of diagnosis information is useful for many problems. [Table 76](#) indicates the minimum set of generally useful diagnostic information that should be collected by software, unless the specifics of the problem are known to be narrower, based on the error code or other information.

It is useful to collect configuration information to ensure that the behavior is not caused by misconfiguration.

**Table 76: Registers Commonly Used for Diagnosis**

MCA Bank	Status	Configuration
MC0	MSR0000_0401 MSR0000_0402 MSR0000_0403	MSR0000_0400 MSRC001_1022 MSRC001_0044
MC1	MSR0000_0405 MSR0000_0406 MSR0000_0407	MSR0000_0404
MC2	MSR0000_0409 MSR0000_040A MSR0000_040B	MSR0000_0408 MSRC001_0046 MSRC001_1023
MC3	Reserved	Reserved
MC4	MSR0000_0411 MSR0000_0412 MSR0000_0413 MSRC000_0408 D18F3x54 D18F2xAC	MSR0000_0410 MSRC001_0048 D18F3x40 D18F3x44 D18F3xE4 D18F3xE8 MSRC001_001F
MC5	MSR0000_0415 MSR0000_0416 MSR0000_0417	MSR0000_0414

If examining MCA registers after startup, determine the cause of the startup:

- INIT; [D18F0x6C\[InitDet\]](#).
- Cold reset; [D18F0x6C\[ColdRstDet\]](#).
- Warm reset; if not INIT or cold reset.

To see if a link failure occurred, examine [D18F0x\[E4,C4,A4,84\]\[LinkFail\]](#). If set, look for additional information:

- Receipt of a sync, such as during a sync flood, saves a status of Sync Error in MC4\_STATUS.
- CRC error saves a status of CRC Error in MC4\_STATUS. See [D18F0x\[E4,C4,A4,84\]\[CrcErr,CrcFloodEn\]](#).
- Link not present does not save status in MC4\_STATUS. See [D18F0x\[E4,C4,A4,84\]\[InitComplete\]](#).

Other registers may be needed depending on the specific error symptoms.

## 2.14.2 DRAM ECC Considerations

DRAM is protected by an error correcting code (ECC). The DRAM error correcting code features an ECC word formed by a symbol based code. The x4 code uses thirty-six 4-bit symbols to make a 144-bit ECC word made up of 128 data bits and 16 check bits.

The x4 code is a single symbol correcting (SSC) and a double symbol detecting (DSD) code. This means the x4 code is able to correct 100% of single symbol errors (any bit error combination within one symbol), and detect 100% of double symbol errors (any bit error combination within two symbols).

### 2.14.2.1 ECC Syndromes

For memory errors, the sections below describe how to find the DIMM in error. The process varies slightly according to the ECC code in use. To determine which ECC code is being used, see [D18F3x180\[EccSymbolSize\]](#).

For correctable errors, the DIMM in error is uniquely identified by the error address ([MSR0000\\_0412\[ErrAddr\]](#)) and the ECC syndrome ([MSR0000\\_0411\[Syndrome\[15:8\]\]](#) and [MSR0000\\_0411\[Syndrome\[7:0\]\]](#)). The error address maps to the two DIMMs composing the 128-bit line, and the ECC syndrome identifies one DIMM by identifying the symbol within the line.

#### 2.14.2.1.1 x4 ECC

The use of x4 ECC is indicated in [D18F3x180\[EccSymbolSize\]](#).

The syndrome field uniquely identifies the failing bit positions of a correctable ECC error. Only syndromes identified by [Table 77](#) 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 77](#), first find the 16-bit syndrome value in the table. This is performed 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, and therefore which DIMM 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 77: x4 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
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	bdfd	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	fed6	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

**Table 77: x4 ECC Correctable Syndromes**

Symbol In Error	Error Bitmask														
	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
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

### 2.14.3 Error Injection and Simulation

*Error injection* allows the introduction of errors into the system for test and debug purposes. See the following sections for error injection details:

- DRAM: See [2.14.3.1 \[DRAM Error Injection\]](#).
- Link:
  - [D18F3x44](#)[GenLinkSel, GenSubLinkSel, GenCrcErrByte1, GenCrcErrByte0].

*Error simulation* involves creating the appearance to software that an error occurred, and can be used to debug machine check interrupt handlers. This is performed by manually setting the MCA registers with desired values, and then driving the software via INT18. See [MSRC001\\_0015](#)[McStatusWrEn] for making MCA registers writable for non-zero values. When McStatusWrEn is set, privileged software can write non-zero values to the specified registers without generating exceptions, and then simulate a machine check using the INT18 instruction (INTn instruction with an operand of 18). Setting a reserved bit in these registers does not generate an exception when this mode is enabled. However, setting a reserved bit may result in undefined behavior.

#### 2.14.3.1 DRAM Error Injection

This section gives details and examples on injecting errors into DRAM using [D18F3xBC\\_x8](#) [DRAM ECC]. The intent of DRAM error injection is to cause a discrepancy between the stored data and the stored ECC value. Therefore, DRAM error injection is only possible on DRAM which supports ECC, and in which [D18F2x90\\_dct\[0\]](#)[DimmEccEn] and [D18F3x44](#)[DramEccEn] are set.

The memory subsystem operates on 64-byte cachelines. The following fields are used to set how the cacheline is to be corrupted in DRAM:

- [D18F3xB8](#)[ArrayAddress] selects a cacheline quadrant (16-byte section) of the cacheline. Each cacheline quadrant is protected by an ECC word. Note that there are special requirements for which bits are used to specify the target quadrant.
- [D18F3xBC\\_x8](#)[ErrInjEn] selects a 16-bit word of the cacheline quadrant selected in ArrayAddress. The 16-bit word identified as ECC[15:0] refers to the bits which store the ECC value; the other 16-bit words address the data on which the ECC is calculated. One or more of these 16-bit words can be selected, and the error bitmask indicated in EccVector is applied to each of the selected words.
- [D18F3xBC\\_x8](#)[EccVector] is a bitmask which selects the individual bits to be corrupted in the 16-bit words selected by ErrInjEn. When selecting the bits to be corrupted for correctable or uncorrectable errors, consider the ECC scheme being used, including symbol size; see [2.14.2 \[DRAM ECC Considerations\]](#) for more details. Note that corrupting more than two symbols may exceed the limits of the ECC to detect the errors; for testing purposes it is recommended that no more than two symbols be corrupted in a single cacheline quadrant.

The distinction between [D18F3xBC\\_x8](#)[DramErrEn] and [D18F3xBC\\_x8](#)[EccWrReq] is that DramErrEn is used to continuously inject errors on every write. This bit is set and cleared by software. EccWrReq is used to inject an error on only one write. This bit is set by software and is cleared by hardware after the error is injected.

When performing DRAM error injection on multi-node systems, [D18F3xB8](#) and [D18F3xBC\\_x8](#) of the NB to which the memory is attached must be programmed.

The following can be used to trigger the injection:

- The memory address is not an explicit parameter of the error injection interface. Once the error injection registers [D18F3xB8](#) and [D18F3xBC](#) are set, the next non-cached access of the appropriate type will trigger the mechanism and apply it to the accessed address. The access should be non-cached so that it is ensured to be seen by the memory controller. Possible methods to ensure a non-cached access include using the appropriate MTRR to set the memory type to UC or turning off caches. If it is important to know the address, then system activity must be quiesced so that the access can take place under careful software control. Once the error injection pattern is set in [D18F3xB8](#) and [D18F3xBC\\_x8](#):
  - Set either [D18F3xBC\\_x8\[EccWrReq\]](#) or [D18F3xBC\\_x8\[DramErrEn\]](#) to enable the triggering mechanism.
  - The next non-cached access of the appropriate type will trigger the mechanism and apply it to the accessed address.

After the error is injected, the data must be accessed in order for the error detection to be triggered. The error address logged in [MSR0000\\_0412](#) will correspond to the cacheline quadrant that contains the error.

When using [MSR0000\\_0411](#) to read `MC4_STATUS` after an error injection and subsequent error detection, be aware that the setting of [D18F3x44\[NbMcaToMstCpuEn\]](#) can cause different cores to see different values. Alternatively, `MC4_STATUS` can be read through the PCI-defined configuration space aliases [D18F3x4C](#) and [D18F3x48](#), which do not return different values to different cores, regardless of the setting of [D18F3x44\[NbMcaToMstCpuEn\]](#).

**Example 1:** Injecting a correctable error:

- Program error pattern:
  - [D18F3xB8\[ArraySelect\]](#)=1000b // select DRAM as target
  - [D18F3xB8\[ArrayAddress\]](#)=000000000b // select 16-byte (128-bit) section
  - [D18F3xBC\\_x8\[ErrInjEn\]](#)=000000001b // select 16-bit word in 16-byte section
  - [D18F3xBC\\_x8\[EccRdReq\]](#)=0 // not a read request
  - [D18F3xBC\\_x8\[EccVector\]](#)=0001h // set bitmask to inject error into only one symbol
- Program error trigger:
  - [D18F3xBC\\_x8\[DramErrEn\]](#)=0 // inject only a single error
  - [D18F3xBC\\_x8\[EccWrReq\]](#)=1 // a write request; enable injection on next write
- Clean up // if programmed for continuous errors
  - [D18F3xBC\\_x8\[DramErrEn\]](#)=0 // inject only a single error



## 2.15 Fusion Controller Hub

The processor contains an integrated Fusion Controller Hub (FCH). The FCH supports the following interfaces:

- Universal Serial Bus (USB) versions 1.1, 2.0, and 3.0
- Serial ATA revision 3.0
- Secure Digital (SD)
- System Management Bus (SMBus)
- Low Pin Count (LPC) bus and SPI interface
- High Definition (HD) audio
- Serial IRQ
- Serial Peripheral Interface (SPI) ROM
- Advanced Configuration and Power Interface (ACPI)

The FCH also comprises the following functions:

- Real-Time Clock (RTC)
- Programmable Interrupt Controller (PIC)
- System clock generation
- System Management Interrupt (SMI)
- General-Purpose I/O (GPIO)
- Power management
- Watchdog Timer (WDT)

### 2.15.1 MMIO Programming for Legacy Devices

The legacy devices LPC, IOAPIC, ACPI and Watchdog Timer require the base address of the Memory Mapped IO registers to be assigned before these logic blocks are accessed. The Memory Mapped IO register base address and its entire range should be mapped to non-posted memory region by programming the CPU register. See BIOS Developer's Guide for details.

### 2.15.2 USB Controllers

The processor supports ten USB ports using the BP\_USB\_HSD[9:0]P/N pins. Ports 0 through 7 are compatible with USB 2.0 (EHCI) and USB 1.1 (OHCI). Ports 8 and 9 are connected to an EHCI/OHCI controller pair or to an xHCI controller, as specified by PMxEF[PortRoutingSelect]. The xHCI controller operates at USB 1.1, USB 2.0, or USB 3.0 speeds. Support for USB 3.0 varies by product.

When ports 8 and 9 are connected to the xHCI controller, these ports are accessed using either the BP\_USB\_HSD[9,8]P/N pins (USB 1.1 and USB 2.0) or the BP\_USB\_SS[1,0][RX,TX]P/N pins (USB 3.0). When ports 8 and 9 are connected to an EHCI/OHCI controller pair, only the BP\_USB\_HSD[9,8]P/N pins are used.

**Table 78: USB Port Mapping**

Ports	PMxEF[PortRoutingSelect]	USB Version	Controller	Registers	Pins
3-0	X	1.1	OHCI 1	D12F0xXX, OHCI1xXX	BP_USB_HSD[3:0]P/N
		2.0	EHCI 1	D12F2xXX, EHCI1xXX	

**Table 78: USB Port Mapping**

Ports	PMxEF[PortRoutingSelect]	USB Version	Controller	Registers	Pins
7-4	X	1.1	OHCI 2	D13F0xXX, OHCI2xXX	BP_USB_HSD[7:4]P/N
		2.0	EHCI 2	D13F2xXX, EHCI2xXX	
9-8	0	1.1	OHCI 3	D16F0xXX, OHCI3xXX	BP_USB_HSD[9,8]P/N
		2.0	EHCI 3	D16F2xXX, EHCI3xXX	
	1	1.1	xHCI	D10F0xXX	BP_USB_HSD[9,8]P/N
2.0					
3.0	BP_USB_SS[1,0][RX,TX]P/N				

Refer to the following for the USB controller register definitions:

- Section 3.25.4.1 [USB 1.1 (OHCI)] for the OHCI controllers

### 2.15.2.1 USB Power Management

USB power management functions are controlled by registers outside the USB controller registers spaces.

See the following register definitions in 3.25.13 [Power Management (PM) Registers]:

- PMx80[UsbPeriodicalSetBmSts, Usb20SetBmSts, Usb11SetBmSts, Usb11BmStsEn]
- PMxED [USB Gating]
- PMxEF [USB Enable]
- PMxF0 [USB Control]

### 2.15.2.2 USB Interrupts

The interrupt mapping is specified by IOC00 [Pci\_Intr\_Index] and IOC01 [Pci\_Intr\_Data]. By default, USB interrupts are routed to PCI INTA#, B#, and C#.

For normal operations, MSI must be disabled in all OHCI and EHCI controllers.

### 2.15.2.3 BIOS Programming Requirements For USB Reset During S3 Resume

Set PMxF0[UsbKbResetEnable]=1 to enable the USB controller to get reset by any software that generates a PCIRST# condition (e.g. a keyboard reset or a write to IOCF9). To avoid losing the USB connection status during the S3 resume procedure, BIOS must ensure that PMxF0[UsbKbResetEnable]=0 before any software-generated PCIRST# reset condition during S3 resume.

### 2.15.2.4 Enabling the xHCI Controller

Software performs the following sequence to enable the xHCI controller:

1. Program PMxEF[Usb3OhciEnable]=0.
2. Program PMxEF[Usb3EhciEnable]=0.
3. Program PMxEF[PortRoutingSelect]=1.
4. Program XHCI\_PMx00[Xhci0Enable]=1.

### 2.15.2.5 OHCI Arbiter Mode

Software should program the following sequence for all OHCI controllers to set OHCI arbiter mode:

1. Whenever the system resumes from S3/S4/S5, software should reset the USB controllers as follows:
  - A. Program PMxD3[AssertUsbRstB]=0.

- B. Wait for <1 ms.
- C. Program `PMxD3[AssertUsbRstB]=1`.
- 2. Program `D[16,13,12]F0x80[OHCIArbiterMode]=11b`.
- 3. Program `D[16,13,12]F0x80[OHCIArbVldCtl]=1b`.

### 2.15.2.6 USB2.0 Controller PHY Configuration and Calibration

Software performs the following sequence to configure EHCI common PHY for each USB port to be used shortly after chip boot up and after USB2.0 EHCI controller initiated the calibration. The register settings have to be re-established to ensure USB2.0 PHY is properly calibrated. It is not needed after waking up from S3.

1. Program `HSSLEW=10b`.
  - A. Program `EHCI[3:1]xB4[PortNumber]` with the controller port number according to [Table 79](#).
  - B. Program `EHCI[3:1]xB4[VControlModeSel]=0110b`.
  - C. Program `HSSLEW` value at `EHCI[3:1]xB4[1:0]=10b`.
  - D. Program `EHCI[3:1]xB4[VLoadB]=1`.
  - E. Wait for `EHCI[3:1]xB4[VBusy]=0`.
  - F. Program `EHCI[3:1]xB4[VLoadB]=0`.
  - G. Wait for `EHCI[3:1]xB4[VBusy]=0`.
  - H. Program `EHCI[3:1]xB4[VLoadB]=1` to lock the PHY control interface.
2. Program `EHCI[3:1]xD0[BgAdj]=6`.
3. Program `EHCI[3:1]xC4[IRefAdj]=2`.
4. Program `EHCI[3:1]xC4[XRefAdj]=2`.
5. Program `EHCI[3:1]xC4[PVI]=1`.
6. Program `EHCI[3:1]xC4[CPAdj]=1`.
7. Program `EHCI[3:1]xC4[DLLControl]=90h`.
8. Program `EHCI[3:1]xD4[PllFilter]=1`.
9. Program `EHCI[3:1]xD4[CalEnable]=0`.
10. Wait for 200 ns.
11. Program `EHCI[3:1]xD4[CalEnable]=1`.
12. Wait for 400 ns.
13. Program `EHCI[3:1]xD4[CalEnable]=0`.

### 2.15.2.7 USB2.0 Controller Common PHY Impedance Calibration

The following sequence enables USB2.0 common PHY impedance calibration.

IF (`D18F3x1FC[USB2CompCal[3:0]]==0`) THEN

- A. Program `EHCI[3:1]xC0[NewCalBus[7:0]]=0`.
- B. Program `EHCI[3:1]xC0[AddToCommonCalibration]=0` and `EHCI[3:1]xC0[UseCommonCalibration]=1` for auto-calibration.

ELSE

- A. Program `EHCI[3:1]xC0[NewCalBus[6:3]]=D18F3x1FC[USB2CompCal[3:0]]` to set the new calibration bus value.
- B. Program `EHCI[3:1]xC0[AddToCommonCalibration]=0` and `EHCI[3:1]xC0[UseCommonCalibration]=0` to have NewCaluBus replace ComCalBus for calibration.

### 2.15.2.8 USB2.0 Controller ISO Device CRC False Error Detection

Software performs the following sequence to program CDR phase shift limit for each USB port to be used:

1. Program `EHCI[3:1]xB4[PortNumber]` with the controller port number according to [Table 79](#).
2. Program `EHCI[3:1]xB4[VControlModeSel]=0111b`.
3. Program `EHCI[3:1]xB4[2:0]=101b`.

4. Program `EHCI[3:1]xB4[VLoadB]=1`.
5. Wait for `EHCI[3:1]xB4[VBusy]=0`.
6. Program `EHCI[3:1]xB4[VLoadB]=0`.
7. Wait for `EHCI[3:1]xB4[VBusy]=0`.
8. Program `EHCI[3:1]xB4[VLoadB]=1` to lock the PHY control interface.

Table 79: USB Port to `EHCI[3:1]xB4[PortNumber]` Mapping

USB Port	Software selects by programming
0	<code>EHCI1xB4[PortNumber]=0h</code>
1	<code>EHCI1xB4[PortNumber]=1h</code>
2	<code>EHCI1xB4[PortNumber]=2h</code>
3	<code>EHCI1xB4[PortNumber]=3h</code>
4	<code>EHCI2xB4[PortNumber]=0h</code>
5	<code>EHCI2xB4[PortNumber]=1h</code>
6	<code>EHCI2xB4[PortNumber]=2h</code>
7	<code>EHCI2xB4[PortNumber]=3h</code>
8	<code>EHCI3xB4[PortNumber]=0h</code>
9	<code>EHCI3xB4[PortNumber]=1h</code>

### 2.15.2.9 xHC USB2.0 Common PHY Calibration

The following programming sequence has to be carried out shortly after chip boot up and after USB2.0 EHCI/XHCI controller initiated the calibration. The register settings have to be re-established to ensure USB2.0 PHY is properly calibrated. It is not needed after waking up from S3.

1. Program `HSSLEW=10b`.
  - A. Program `D10F0x4C_x4000_0000[PortNumber]` with the controller port number according to Table 80.
  - B. Program `D10F0x4C_x4000_0000[VControlModeSel]=0110b`.
  - C. Program `HSSLEW` value at `D10F0x4C_x4000_0000[1:0]=10b`.
  - D. Program `D10F0x4C_x4000_0000[VLoadB]=1`.
  - E. Wait for `D10F0x4C_x4000_0000[VBusy]=0`.
  - F. Program `D10F0x4C_x4000_0000[VLoadB]=0`.
  - G. Wait for `D10F0x4C_x4000_0000[VBusy]=0`.
  - H. Program `D10F0x4C_x4000_0000[VLoadB]=1` to lock the PHY control interface.
2. Program `D10F0x4C_x4000_0050[BgAdj]=6`.
3. Program `D10F0x4C_x4000_000C[IRefAdj]=2`.
4. Program `D10F0x4C_x4000_000C[XRefAdj]=2`.
5. Program `D10F0x4C_x4000_000C[PVI]=1`.
6. Program `D10F0x4C_x4000_000C[CPAdj]=1`.
7. Program `D10F0x4C_x4000_000C[DllControl]=90h`.
8. Program `D10F0x4C_x4000_0054[PllFilter]=1`.
9. Program `D10F0x4C_x4000_0054[CalEnable]=0`.
10. Wait for 200 ns.
11. Program `D10F0x4C_x4000_0054[CalEnable]=1`.
12. Wait for 400 ns.
13. Program `D10F0x4C_x4000_0054[CalEnable]=0`.

### 2.15.2.10 USB3.0 PHY Auto-Calibration Enablement

BIOS enables USB3.0 PHY auto-calibration as follows:

1. Program `XHCI_PMx8C[PIIVcoTune[3:0]]=0011b`.
2. Program `XHCI_PMx8C[CrPllCalibEn]=1`.

### 2.15.2.11 xHCI ISO Device CRC False Error Detection

Software performs the following sequence to program CDR phase shift limit for each USB port to be used:

1. Program `D10F0x4C_x4000_0000[PortNumber]` with the controller port number according to [Table 80](#).
2. Program `D10F0x4C_x4000_0000[VControlModeSel]=0111b`.
3. Program `D10F0x4C_x4000_0000[2:0]=101b`.
4. Program `D10F0x4C_x4000_0000[VLoadB]=1`.
5. Wait for `D10F0x4C_x4000_0000[VBusy]=0`.
6. Program `D10F0x4C_x4000_0000[VLoadB]=0`.
7. Wait for `D10F0x4C_x4000_0000[VBusy]=0`.
8. Program `D10F0x4C_x4000_0000[VLoadB]=1` to lock the PHY control interface.

Table 80: USB Port to `D10F0x4C_x4000_0000[PortNumber]` Mapping

USB Port	Software selects by programming
8	<code>D10F0x4C_x4000_0000[PortNumber]=0h</code>
9	<code>D10F0x4C_x4000_0000[PortNumber]=1h</code>

### 2.15.2.12 xHCI PHY Clock Gating

The following programming sequence configures the clock gating in the xHCI PHY. They must be performed for each port after enabling xHCI controller and need to be restored after all power state resumes.

1. Program `D10F0x4C_x4000_0000[VLoadB]=1`.
2. Program `D10F0x4C_x4000_0000[PortNumber]` with the controller port number according to [Table 80](#).
3. Program `D10F0x4C_x4000_0000[VControlModeSel]=0011b`.
4. Program `D10F0x4C_x4000_0000[2]=1`.
5. Read `D10F0x4C_x4000_0000[VBusy]` to ensure it is 0.
6. Program `D10F0x4C_x4000_0000[VLoadB]=0`.
7. Wait for `D10F0x4C_x4000_0000[VBusy]=0`.
8. Program `D10F0x4C_x4000_0000[VLoadB]=1` to lock the PHY control interface.

### 2.15.2.13 xHCI Firmware Preload

Software performs the following programming sequence for firmware preload. Firmware preload should be executed before BIOS de-asserts `XHCI_PMx00[U3CoreReset]`.

1. Enable firmware preload by programming `XHCI_PMx00[FwLoadMode]=1`.
2. Program `XHCI_PMx04[XhciFwPreloadType]` with 1 for boot RAM preload and 0 for instruction RAM preload.
3. Program `XHCI_PMx04[XhciFwRomAddr]` with the offset of the application firmware in the external ROM.
4. Program `XHCI_PMx08[XhciFwRamAddr]= 0`.
5. Program `XHCI_PMx08[XhciFwSize]=8000h`.
6. Program `XHCI_PMx00[FwPreloadStart]=1` to start firmware preload.
7. Wait for `XHCI_PMx00[FwPreloadComplete]=1`.
8. Program `XHCI_PMx00[FwPreloadStart]=0`.

9. Program `XHCI_PMx00[U3CoreReset]=0` to de-assert xHC reset.

#### 2.15.2.14 xHCI Clear Pending PME on Sx State Entry

During S3/S4 entry with wake from Sx state enabled, if the xHCI controller received a wake event before the system shutdown into Sx state is completed, `D10F0x54[PmeStatus]` may remain set when the system enters into Sx state. This will prevent subsequent wake events from being propagated to the ACPI controller. BIOS should write 1 to clear `D10F0x54[PmeStatus]` if `D10F0x54[PmeStatus]` is 1 and ACPI GEvent status bit is clear on entry into S3/S4 state.

#### 2.15.2.15 xHCI Enable OHCI3/EHCI3 on S4/S5 State Entry

If xHCI controller is enabled, OHCI3/EHCI3 controllers are disabled for normal operations. See [2.15.2.4 \[Enabling the xHCI Controller\]](#). On S4/S5 entry, OHCI3/EHCI3 controllers should be enabled .

- Program `PMxEF[Usb3OhciEnable]=1`.
- Program `PMxEF[Usb3EhciEnable]=1`.

### 2.15.3 SATA

#### 2.15.3.1 SATA Operating Mode

The SATA host controller can operate in the following modes:

- IDE mode.
- AHCI mode.

Software programs the subclass code and programming interface register to enable the SATA controller as IDE controller or AHCI controller.

1. Program `D11F0x40[SubclassCodeWriteEnable]=1`.
2. Program `D11F0x08[SubclassCode]` and `D11F0x08[ProgramIF[7:0]]` according to [Table 231 \[SATA Controller Subclass Code and ProgramIF Settings\]](#).
3. Program `D11F0x40[SubclassCodeWriteEnable]=0`.

#### 2.15.3.2 SATA Drive Detection in IDE mode

The following sequence should be included in the SBIOS drive identification loop for SATA drive detection in IDE mode.

1. If any of the SATA port status register at `SATAx1[A,2]8[DET]==03h`, program `IDE[1:0]x06[DriveHead]=A0h` for the corresponding port and go to Step 2.  
Else, no SATA drives are attached; exit the detection loop program.
2. If `(IDE[1:0]x06[DriveHead]==A0h) && (IDE[1:0]x07[7]==0) && (IDE[1:0]x07[3]==0)` for a port, the SATA device on that port is ready.  
Else, loop until 30 s time-out; There is no SATA device attached if a time-out occurs.  
Note: Most drives don't need 30 s time-out. The 30 s time-out is only needed for some particularly large capacity SATA drives, which require a longer spin-up time during a cold boot.

#### 2.15.3.3 SATA PHY Auto-Calibration Enablement

BIOS enables SATA PHY auto-calibration as follows:

1. Program `D11F0x88[PIIVcoTune]=0011b`.
2. Program `PMxDC[PIICalibEn]=1`.
3. Reset PHY:

- A. Program `D11F0x84[RSTB]=0` to assert reset.
- B. Wait for 100 us.
- C. Program `D11F0x84[RSTB]=1` to release PHY reset.

#### 2.15.3.4 SATA PHY Fine Tuning

The SBIOS should program the SATA controller in the sequence indicated below to fine tune the PHY. Performing this procedure provides sufficient time for the SATA controllers to correctly complete SATA drive detection. The same procedure is required after the system resumes from the S3 state.

- Gen 3 settings:
    - Program `D11F0x80[15:0]=0130h` to select Gen 3 for all ports.
    - Program `D11F0x98[31:0]=0040f407h`.
  - Gen 2 settings:
    - Program `D11F0x80[15:0]=0120h` to select Gen 2 for all ports.
    - Program `D11F0x98[31:0]=00403204h` to fine tune PHY for Gen 2.
  - Gen 1 settings:
    - Program `D11F0x80[15:0]=0110h` to select Gen 1 for all ports.
    - Program `D11F0x98[31:0]=00403103h` to fine tune PHY for Gen 1.
  - Squelch detector settings:
    - A. `PMxC8[EPromDeSelect]=1b`.
    - B. Program `PMxD8[EpromIndex]=01h`.
    - C. Read from `PMxD9[4]`.
    - D. IF (`PMxD9[4]==0`) THEN
      - , program the squelch detector threshold for all ports at all generation speed as follows:
        - Program `D11F0x80[15:0]=0130h`.
        - Program `D11F0x9C[RX_SQDET_TH]=101b`.
        - Program `D11F0x80[15:0]=0120h`.
        - Program `D11F0x9C[RX_SQDET_TH]=101b`.
        - Program `D11F0x80[15:0]=0110h`.
        - Program `D11F0x9C[RX_SQDET_TH]=101b`.
      - ELSE
        - a. `PMxD8[EpromIndex]=02h`.
        - b. Read from `PMxD9[2:0]`.
        - c. Program the squelch detector threshold control register for port 1 at all generation speed:
          - Program `D11F0x80[15:0]=0031h`.
          - Program `D11F0x9C[RX_SQDET_TH]=PMxD9[2:0]`.
          - Program `D11F0x80[15:0]=0021h`.
          - Program `D11F0x9C[RX_SQDET_TH]=PMxD9[2:0]`.
          - Program `D11F0x80[15:0]=0011h`.
          - Program `D11F0x9C[RX_SQDET_TH]=PMxD9[2:0]`.
        - d. `PMxD8[EpromIndex]=01h`.
        - e. Read from `PMxD9[7:5]`.
        - f. Program the squelch detector threshold control register for port 0 at all generation speed:
          - Program `D11F0x80[15:0]=0030h`.
          - Program `D11F0x9C[RX_SQDET_TH]=PMxD9[7:5]`.
          - Program `D11F0x80[15:0]=0020h`.
          - Program `D11F0x9C[RX_SQDET_TH]=PMxD9[7:5]`.
          - Program `D11F0x80[15:0]=0010h`.
          - Program `D11F0x9C[RX_SQDET_TH]=PMxD9[7:5]`.
- ENDIF.

### 2.15.3.5 SATA PHY Reference Clock Selection

SATA PHY reference clock can originate from either internal or external clock. The following steps select SATA PHY reference clock source:

1. Select one of the following options for reference clock.
  - Internal 48 MHz differential non-spread clock from pad (default):
    - a. Program `PMxDA[RefClkSel]=1` to select reference clock from internal RDL.
    - b. Program `PMxDA[RefDivSel]=0` to set PHY divider to div-by-1.
    - c. Program `D11F0x8C[PLL_CLKF]=7Dh` to set PLL feedback clock divider value to 6 GHz/48 MHz=7Dh.
  - External 25 MHz crystal non-spread clock from external clock chip (the clock path has to be routed on the board, in order for the setting to take effect):
    - a. Program `PMxDA[RefClkSel]=0` to select reference clock from an external source, via the PAD\_XTALI and PAD\_XTALO, to SATA PHY.
    - b. Program `PMxDA[RefDivSel]=0` to set PHY divider to div-by-1.
    - c. Program `D11F0x8C[PLL_CLKF]=F0h` to set PLL feedback clock divider value to 6 GHz/25 MHz=F0h.
2. Reset PHY:
  - A. Program `D11F0x84[RSTB]=0` to assert reset.
  - B. Wait for 100 us.
  - C. Program `D11F0x84[RSTB]=1` to release PHY reset.

### 2.15.3.6 SATA Power Management

#### 2.15.3.6.1 SATA PHY Power Saving

The SATA PHY has different power saving states with the Active state consuming the most power. The table below summarizes the different controls during each power saving state.

**Table 81: SATA PHY Power Saving States**

Power Saving States	<code>PMxDA[SetMaxGen2]</code>	<code>PMxDA[SataEnable]</code>	<code>SATAx1[9,1]8[ICC]</code>	<code>D11F0x40[SataPortDisable]</code>
Active	0	1	1	0
Active with SetMaxGen2	1	1	1	0
Partial	x	1	2	0
Slumber	x	1	6	0
Lane Disabled	x	1	x	1
PHY Power Down	x	0	x	1

#### 2.15.3.7 Enable Shadow Register Reload

When default PHY settings are not optimal, they need to be adjusted by software. Once these settings are re-programmed, software shall program `D11F0x84[S5ShadowLdDis]` to 0 to turn on shadow register reloading. When default PHY settings are optimal and need no adjustment, `D11F0x84[S5ShadowLdDis]` needs not to be altered and shadow register preloading is effectively disabled.



### 2.15.3.8 SATA Interrupt Handling

#### 2.15.3.8.1 Line Interrupt

The SATA interrupt mapping is specified by [IOC00 \[Pci\\_Intr\\_Index\]](#) and [IOC01 \[Pci\\_Intr\\_Data\]](#).

#### 2.15.3.8.2 MSI Message

SATA controller supports message-based interrupts. When MSI is enabled and the SATA controller owns N ports, as specified by [SATAx0C \[Ports Implemented \(PI\)\]](#), [D11F0x50\[MMC\]](#) should be programmed according to platform configurations:

- If [SATAx00\[CCCS\]](#)==1, [D11F0x50\[MMC\]](#) should be the minimum of 2M which satisfies  $2M \geq (N+1)$ . For example: if BIOS knows that system has 2 SATA ports in AHCI mode (N=2). If [SATAx00\[CCCS\]](#)==1, BIOS should program [D11F0x50\[MMC\]](#)=2 to request 4 MSI interrupts.

MSI capability pointer should be hidden when SATA subclass is configured in IDE mode (see [2.15.3.1 \[SATA Operating Mode\]](#)).

#### 2.15.3.9 Clear status of SATA PERR

BIOS should clear SATA PERR status soon after a cold boot, warm boot, any S3/S1 resume events, or an [IOCF9](#) initiated reset outside a warm boot before any SATA activity is initiated.

- Write 1 to clear [SMIx3C\[SataPerr\]](#).
- Write 1 to clear [SMIx84\[RasEvent55\]](#).

### 2.15.4 LPC Bus Interface

BIOS should program [D14F3xA0\[SpibaseAddr\]](#) with non-zero address to enable the MMIO access to SPI ROM control registers.

#### 2.15.4.1 Enabling LPC DMA function

If DMA is required for the LPC interface, program the registers as the following:

- Program [D14F3x40\[LegacyDmaEnable\]](#)=1.
- Program [D14F3x78\[NoHog\]](#)=1.
- Program [PMx08\[ArbDmaDis\]](#)=1.
- Program [D14F3x78\[LDRQ1\]](#)=1 and [D14F3x78\[LDRQ0\]](#)=1.
- Program [PMx04\[LegacyDmaPrefetchEnhance\]](#)=1 for non-DOS mode. Note: This bit should only be enabled in the ACPI method (called by the OS). This ensures that it is enabled only when the system is in Windows® mode. Under DOS mode, this feature may not work properly and may cause the floppy to malfunction.

#### 2.15.4.2 Enabling SPI 100

To enable the support for SPI 100 MHz speed, software needs to program [SPIx20\[UseSpi100\]](#)=1. SPI 100 should be enabled on after the Auto ROM sizing has been completed (Auto ROM sizing is done by the software only during the initial boot when the system power state transitions from G3->S5->S0). The actual read speed also depends on the settings at [SPIx22 \[SPI100 Speed Config\]](#).

### 2.15.5 SD Controller

The SD controller is SD 3.0 compliant host controller.

### 2.15.6 HD Audio

The FCH has one High Definition Audio controller (HD audio), which can be enabled/disabled by programming [PMxEB\[AzEnable\]](#).

The HD audio controller supports up to 4 codecs with one AZ\_SDIN pin from each codec. The four AZ\_SDIN pins are multiplexed with GPIO167-170 (GPIOxA7-GPIOxAA). If a particular pin is to be used for HD audio functionality and the integrated pull-down is to be used rather than external pull-down resistor, in addition to being configured for HD audio, the following bits need to be set:

- GPIOxA7=3Eh
- GPIOxA8=3Eh
- GPIOxA9=3Eh
- GPIOxAA=3Eh

For example, if only GPIO167 and GPIO168 are to be used for HD audio, then only GPIOxA7 and GPIOxA8 need to be programmed to 3Eh. See [3.25.12 \[GPIO Pin control registers\]](#).

#### 2.15.6.1 HD Audio AF and MSI Capability

The Advanced Feature (AF) capability and MSI capability in HD audio controller are controlled by [D14F2x44\[EnAFCap\]](#) and [D14F2x44\[EnMsiCap\]](#) respectively. The settings of these two bits affect how the AF capability and MSI capability are linked to the capability pointer linked list.

**Table 82: HD Audio AF and MSI Capability Settings**

<a href="#">D14F2x44</a> [EnAF-Cap]	<a href="#">D14F2x44</a> [EnMsi-Cap]	<a href="#">D14F2x50</a> [NextPtr]	<a href="#">D14F2x60</a> [CapID]	<a href="#">D14F2x60</a> [NextPtr]	<a href="#">D14F2x70</a> [AFC-apID]	<a href="#">D14F2x70</a> [NextPtr]	<a href="#">D14F2x70</a> [AFLengt h]	<a href="#">D14F2x70</a> [25:24]
0	0	0h	0h	0h	0h	0h	0h	0h
0	1	60h	5h	0h	0h	0h	0h	0h
1	0	70h	0h	0h	13h	0h	6h	3h
1	1	60h	5h	70h	13h	0h	6h	3h

### 2.15.7 ASF Controller

The ASF controller can be used in a slave mode to support DASH clients. When the platform is configured for DASH the BIOS should at the minimum set the following registers to ensure the ASF is configured in slave mode and the control commands are defined properly in the ASF tables.

- When operating in Master mode, BIOS should program [PMx28\[AsfSmMasterEn\]](#)=1. BIOS should program [PMx28\[AsfSmMasterEn\]](#) back to 0 after the master operations are done.
- When operating in Slave mode, BIOS should program [PMx28\[AsfSmMasterEn\]](#)=0.

In addition to support remote DASH operations, BIOS should ensure that the ASF tables define the correct command codes and [ASFx0E \[RemoteCtrlAdr\]](#) has to be programmed with the same value reported in the ASF control data table.

ASF controller can behave like remote control device to accept the command from client side to do the reset/power down/power up/power cycle. BIOS reports those in ASF control data table. The command and command data are defined in [Table 83](#).

**Table 83: ASF Remote Control Commands**

Remote Control	Control Command	Control data
Reset	50h	00h
Power Up	51h	00h
Power Down	52h	00h
Power Cycle	53h	00h

### 2.15.8 Integrated Micro-Controller

The FCH supports an integrated micro-controller (IMC) .

The IMC can be enabled by programming [PMxD6\[ImcEnable\]](#).

### 2.15.9 On-Chip Clock Generator

There are two clocking modes in which this chip can be brought up based on the boot strap pin LPCCLK1. SBIOS should read [MISCx80\[ClkGenStrap\]](#) to determine clock generator mode. SBIOS should program [MISCx40\[OscClkSwitchEn\]](#) to 1 to select average 14 MHz OSC clock provided by internal PLL in both internal and external clocking mode.

#### 2.15.9.1 Power Saving In Internal Clock Mode

GPP\_CLK\_P/N pins are powered off when this chip is strapped to use an external clock, and powered on when strapped to operate in internal clock mode. The GPP\_CLK clocks are mapped to corresponding CLK\_REQ# pins. The GPP\_CLK0/1/2/3 mapping is defined at [MISCx00\[15:0\]](#). SLT\_GFX\_CLK mapping is defined at [MISCx04\[7:4\]](#). In internal clock mode, a selected GPP\_CLK can be powered off when the corresponding CLK\_REQ# is asserted. See [MISCx00](#) && [MISCx04](#) for details.

Software can also program [MISCx04\[PCIE\\_RCLK\\_PowerDownEnable\]=1](#) to turn off 100 MHz reference clock input buffer in internal clock generator mode for power saving.

#### 2.15.9.2 Global A-Link / B-Link Clock Gating

Software needs to program [MISCx2C\[AlinkClkGateOffEn\]=1](#) and [PMx04\[ABLinkClkGateEn\]=1](#) to enable global A-link clock gate off function.

Software needs to program [MISCx2C\[BlinkClkGateOffEn\]=1](#) and [PMx04\[ABLinkClkGateEn\]=1](#) to enable global B-link clock gate off function.

[PMx04\[ABLinkClkGateEn\]](#) is a non-sticky bit and needs to be programmed to 1 after PCI reset and S3/S4/S5 state, if global A-link / B-link gating function has been enabled. [MISCx2C\[AlinkClkGateOffEn\]](#) and [MISCx2C\[BlinkClkGateOffEn\]](#) are sticky bits.

#### 2.15.9.3 CG\_PLL CMOS Clock Driver Setting for Power Saving

Software programs the following registers to select CMOS clock driver for CG1\_PLL in internal & external clock mode for power saving:

- A. Program [MISCx1C\[CgpllClkDriverUpdate\]=0](#).
- B. Program [MISCx1C\[Cg1ClkDriverType\]=1011b](#) and Program [MISCx1C\[Cg1RefClk48MHzDriverType\]=0010b](#).
- C. Perform a software reset through [IOCF9](#).

These settings will only need to be applied once when the chip is booting up from G3-S5->S0. These is no

need to do this for every reset since the setting is sticky.

### 2.15.10 FCH Gasket

The FCH gasket (SBG) is responsible for converting between the protocols used by the interface and the A-link bridge interface. BIOS should program [PMxE0](#) to set up base address before accessing AB configuration registers.

### 2.15.11 A-Link Bridge

The A-link bridge (AB) sits behind the FCH gasket.

#### 2.15.11.1 Detection of Upstream Interrupts

BIOS should enable AB to detect upstream interrupts for the purpose of system management.

- Program [ABx04\\_x94](#)[MsiAddr[39:20]] with CPU interrupt delivery address [39:20].
- Program [ABx04\\_x94](#)[MsiAddrEn]=1.

#### 2.15.11.2 AB Memory Power Saving

The following sequence enables AB memory power saving:

- A. Program [MISCx68](#)[ABBypassMemDsd]=0 to enable AB memory DSD.
- B. Program [ABx04\\_x58](#)[BIMemSDEn]=1 to enable B-link memory shutdown.
- C. Program [ABx04\\_x58](#)[AIMemSDEn]=1 to enable A-link memory shutdown.

#### 2.15.11.3 AB Internal Clock Gating

The following sequence enables AB internal clock gating:

- A. Program [ABx04\\_x54](#)[BIClkGateDelay]=10h to set the number of cycles to delay before gating B-link clocks.
- B. Program [ABx04\\_x10054](#)[AIClkGateDelay]=10h to set the number of cycles to delay before gating A-link clocks.
- C. Program [ABx04\\_x54](#)[BIClkGateEn]=1 to enable medium grain B-link clock gating.
- D. Program [ABx04\\_x54](#)[DbgClkGateEn]=1 to disable debug only B-link clocks.
- E. Program [ABx04\\_x10054](#)[AIClkGateEn]=1 to enable medium grain A-link clock gating.
- F. Program [ABx04\\_x10054](#)[DbgClkGateEn]=1 to disable debug only A-link clocks.

#### 2.15.11.4 AB 32/64 Byte DMA Write Enable

To enable AB 32 byte/64 byte DMA write, software needs to program [ABx04\\_x54](#)[UpWr16BMode]=0 and [ABx04\\_x54](#)[UpSWrByteCntSbgMode]=1 to disable 16 byte write splitting and enable SBG mode byte-count calculation logic. Software should also make sure [ABx04\\_x204](#)[Dma16ByteMode]=0.

### 3 Registers

This section provides detailed field definitions for the core 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, [IOCF8 \[IO-Space Configuration Address\]](#) and [IOCFC \[IO-Space Configuration Data Port\]](#), and the legacy block configuration registers. Unless otherwise specified, there is one set of these registers per node; the registers in a node are accessible to any core on that node. See [3.2 \[IO Space Registers\]](#) and [3.25.1 \[Legacy Block Configuration Registers \(IO\)\]](#).
- **APICXX0**: APIC memory-mapped registers; XX0 is the hexadecimal byte address offset from the base address. See [2.4.8.1.2 \[APIC Register Space\]](#).
- **CPUID FnXXXX\_XXXX\_EiX[\_xYYY]**: processor capabilities information returned by the CPUID instruction. See [3.17 \[CPUID Instruction Registers\]](#). Each core may only access this information for itself.
- **MSRXXXX\_XXXX**: **MSRs**; XXXX\_XXXX is the hexadecimal MSR number. This space is accessed through x86-defined RDMSR and WRMSR instructions. Unless otherwise specified there is one set of these registers **Per-core**. See [2.4.1 \[L2 complex\]](#).
- **DXFYxZZZ**: PCI-defined configuration space; X specifies the hexadecimal device number (this may be 1 or 2 digits), Y specifies the function number, and ZZZ specifies the hexadecimal byte address (this may be 2 or 3 digits); e.g., D18F3x40 specifies the register at device 18h, function 3, and address 40h. See [2.7 \[Configuration Space\]](#), for details about configuration space.
  - Some register in D18F2xXXX have the `_dct[0]` mnemonic suffix. See [2.9.3 \[DCT Configuration Registers\]](#).
- **PMCxXXX**: core performance monitor events; XXX is the hexadecimal event counter number programmed into [MSRC001\\_00\[03:00\]\[EventSelect\]](#); See [2.6.1.1 \[Core Performance Monitor Counters\]](#).
  - When PMCxXXX is followed by [z:y] then UnitMask[z:y] is being specified.
- **L2IPMCxXXX**: L2 performance monitor events; XXX is the hexadecimal event counter number programmed into [MSRC001\\_023\[6,4,2,0\]\[EventSelect\]](#); See [2.6.1.2 \[L2I Performance Monitor Counters\]](#).
  - When PMCxXXX is followed by [z:y] then UnitMask[z:y] is being specified.
- **NBPMCxXXX**: NB performance monitor events; XXX is the hexadecimal event counter number programmed into [MSRC001\\_024\[6,4,2,0\]\[EventSelect\]](#); See [2.6.1.3 \[NB Performance Monitor Counters\]](#).
  - When NBPMCxXXX is followed by [z:y] then UnitMask[z:y] is being specified.
- **ABxXX**: ALink Bridge registers; XX specifies the hexadecimal byte address offset from the base address. The base address for this space is specified by [PMxE0 \[ABRegBar\]](#). Unless otherwise specified, there is one set of these registers per node; the registers in a node are accessible to any core on that node. See [3.25.2 \[AB Configuration Registers\]](#).
- **IDE[Y]xXX**: SATA controller IDE mode IO mapped registers; XX specifies the hexadecimal byte address offset from the base address; Y specifies primary drive (0) or secondary drive (1). The base address for this space is specified in [Table 232 \[IDE Compatibility Mode and Native Mode Address Mapping\]](#). Unless otherwise specified, there is one set of these registers per node; the registers in a node are accessible to any core on that node. See [3.25.3.2.1 \[IDE Compatibility Mode and Native Mode Registers\]](#).
- **IDE\_BMxXX**: SATA controller IDE mode IO mapped bus master registers; XX specifies the hexadecimal byte address offset from the base address. The base address for this space is specified by [D11F0x20 \[Bus Master Interface Register Base Address \(BAR4\)\]](#). Unless otherwise specified, there is one set of these registers per node; the registers in a node are accessible to any core on that node. See [3.25.3.2.2 \[IDE Bus Master Registers\]](#).
- **SATAxXXX**: SATA controller AHCI mode memory mapped registers; XXX specifies the hexadecimal byte

address offset from the base address. The base address for this space is specified by [D11F0x24 \[AHCI Base Address \(BAR5\)\]](#). Unless otherwise specified, there is one set of these registers per node; the registers in a node are accessible to any core on that node. See [3.25.3.3 \[SATA Memory Mapped AHCI Registers\]](#).

- **SATA\_EMxXX**: SATA controller AHCI mode memory mapped enclosure buffer management registers; XX specifies the hexadecimal byte address offset. Unless otherwise specified, there is one set of these registers per node; the registers in a node are accessible to any core on that node. See [3.25.3.3.3 \[Enclosure Buffer Management Registers\]](#).
- **OHCI[Y]xXXX**: USB OHCI controller memory mapped registers; XXX specifies the hexadecimal byte address offset from the base address; Y specifies the OHCI controller number; e.g., OHCI[1]x04 specifies OHCI controller 1 memory mapped control register offset 04. The base address for this space is specified by [D\[16,13,12\]F0x10 \[OHCI Base Address\]](#). Unless otherwise specified, there is one set of these registers per node; the registers in a node are accessible to any core on that node. See [3.25.4.1.2 \[OHCI Memory Mapped IO Registers\]](#).
- **EHCI[Y]xXX**: USB EHCI controller memory mapped registers; XX specifies the hexadecimal byte address offset from the base address; Y specifies the EHCI controller number; e.g., EHCI[1]x04 specifies EHCI controller 1 memory mapped control register offset 04. The base address for this space is specified by [D\[16,13,12\]F2x10 \[BAR\\_EHCI\]](#). Unless otherwise specified, there is one set of these registers per node; the registers in a node are accessible to any core on that node. See [3.25.4.2.2 \[EHCI Memory Mapped IO Registers\]](#).
- **xHCI\_PMxXX**: USB xHCI controller ACPI memory mapped registers; XX specifies the hexadecimal byte address offset from the base address. Unless otherwise specified, there is one set of these registers per node; the registers in a node are accessible to any core on that node. See [3.25.4.3.2 \[xHCI Power Management Registers\]](#).
- **SDHCxXX**: Secure Digital host controller memory mapped registers; XX specifies the hexadecimal byte address offset from the base address. The base address for this space is specified by {[D14F7x14 \[Upper Base Address Reg 0\]](#),[D14F7x10 \[Base Address Reg 0\]](#)}. Unless otherwise specified, there is one set of these registers per node; the registers in a node are accessible to any core on that node. See [3.25.6.2 \[SD Host Controller Configuration Registers \(SDHC\)\]](#).
- **ASFxXX**: ASF registers; XX specifies the hexadecimal byte address offset. Unless otherwise specified, there is one set of these registers per node; the registers in a node are accessible to any core on that node. See [3.25.7.2 \[ASF \(Alert Standard Format\) Registers\]](#).
- **SMBUSxXX**: SMBus registers; XX specifies the hexadecimal byte address offset. Unless otherwise specified, there is one set of these registers per node; the registers in a node are accessible to any core on that node. See [3.25.7.3 \[SMBus Registers\]](#).
- **IOAPICxXX**: IOAPIC registers; XX specifies the hexadecimal byte address offset from the base address. Unless otherwise specified, there is one set of these registers per node; the registers in a node are accessible to any core on that node. See [3.25.8 \[IOAPIC Registers\]](#).
- **SPIxXX**: SPI memory mapped registers; XX specifies the hexadecimal byte address offset from the base address. The base address is specified by [D14F3xA0 \[SPI Base\\_Addr\]](#). Unless otherwise specified, there is one set of these registers per node; the registers in a node are accessible to any core on that node. See [3.25.9.2 \[SPI Registers\]](#).
- **HPETxXXX**: HPET registers; XXX specifies the hexadecimal byte address offset. Unless otherwise specified, there is one set of these registers per node; the registers in a node are accessible to any core on that node. See [3.25.10 \[High Precision Event Timer \(HPET\) Registers\]](#).
- **MISCxXX**: ACPI miscellaneous control registers; XX specifies the hexadecimal byte address offset. Unless otherwise specified, there is one set of these registers per node; the registers in a node are accessible to any core on that node. See [3.25.11 \[Miscellaneous \(MISC\) Registers\]](#).
- **GPIOxXX**: GPIO registers; XX specifies the hexadecimal byte address offset. Unless otherwise specified, there is one set of these registers per node; the registers in a node are accessible to any core on that node. See [3.25.12.1 \[GPIO Registers\]](#).
- **IOMUXxXX**: IOMUX registers; XX specifies the hexadecimal byte address offset. Unless otherwise speci-

fied, there is one set of these registers per node; the registers in a node are accessible to any core on that node. See 3.25.12.2 [IOMux Registers].

- **PMxXX**: power management registers; XX specifies the hexadecimal byte address offset. Unless otherwise specified, there is one set of these registers per node; the registers in a node are accessible to any core on that node. See 3.25.13 [Power Management (PM) Registers].
- **PM2xXX**: power management block 2 registers; XX specifies the hexadecimal byte address offset. Unless otherwise specified, there is one set of these registers per node; the registers in a node are accessible to any core on that node. See 3.25.14 [Power Management Block 2 (PM2) Registers].
- **SMIxXX**: SMI registers; XX specifies the hexadecimal byte address offset. Unless otherwise specified, there is one set of these registers per node; the registers in a node are accessible to any core on that node. See 3.25.16 [SMI Registers].
- **WDTxXX**: Watchdog timer registers; XX specifies the hexadecimal byte address offset. Unless otherwise specified, there is one set of these registers per node; the registers in a node are accessible to any core on that node. See 3.25.17 [Watchdog Timer (WDT) Registers].
- **AcDcTimerxXX**: AC/DC wake alarm timer registers; XX specifies the hexadecimal byte address offset. Unless otherwise specified, there is one set of these registers per node; the registers in a node are accessible to any core on that node. See 3.25.18 [Wake Alarm Device (AcDcTimer) Registers].

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 an explicit step [A:BstepC]: Defines the registers from A to B, 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 processor 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 84: Terminology in Register Descriptions**

Term	Definition
<b>BIOS</b>	Software recommendation syntax. See 3.1.2 [Software Recommendation (BIOS, SBIOS)].
<b>SBIOS</b>	
<b>See</b>	Reference to remote definition.
<b>Alias</b>	The alias keyword allows the definition of a soft link between two registers. <ul style="list-style-type: none"> <li>• X is an alias of Y: X is a soft link to the register Y.</li> <li>• X1, X2 are an alias of Y: Both X1 and X2 are soft links to Y.</li> </ul>

**Table 84: Terminology in Register Descriptions**

Term	Definition
<b>IF</b>	<p>Allows conditional definition as a function of register fields. The syntax is:</p> <ul style="list-style-type: none"> <li>• IF (conditional-expression) THEN definition ENDIF.</li> <li>• IF (conditional-expression) THEN definition ELSE definition ENDIF.</li> <li>• IF (conditional-expression) THEN definition ELSEIF (conditional-expression) THEN definition ELSE definition ENDIF.</li> </ul>
<b>THEN</b>	
<b>ELSEIF</b>	
<b>ELSE</b>	
<b>ENDIF</b>	
<b>Access Types</b>	
<b>Read</b>	Capable of being read by software.
<b>Read-only</b>	Capable of being read but not written by software.
<b>Write</b>	Capable of being written by software.
<b>Write-only</b>	Write-only. Capable of being written by software. Reads are undefined.
<b>Read-write</b>	Capable of being written by software and read by software.
<b>Set-by-hardware</b>	Register field is set high by hardware, set low by hardware, or updated by hardware.
<b>Cleared-by-hardware</b>	
<b>Updated-by-hardware</b>	
<b>Updated-by-SMU</b>	
<b>Write-1-to-clear</b>	Software must write a 1 to the bit in order to clear it. Writing a 0 to these bits has no affect.
<b>Write-1-only</b>	Software can set the bit high by writing a 1 to it. Writes of 0 have no effect.
<b>Reset-applied</b>	Takes effect on warm reset.
<b>GP-read</b>	GP exception occurs on read.
<b>GP-write</b>	GP exception occurs on write.
<b>GP-read-write</b>	GP exception occurs on a read or a write.
<b>Per-core</b>	One instance per core. Only valid for MMIO config space. Writes of these bits from one core only affect that core's register. Reads return the values appropriate to that core.
<b>Per-compute-unit</b>	One instance per compute unit. Writes of these bits from one core only affect that compute unit's register. Reads return the values appropriate to that compute unit. See 2.4.2.1 [Registers Shared by Cores in a L2 complex].
<b>Per-L2</b>	One instance per L2 cache. See <a href="#">CUID Fn8000_001D_EAX_x2[NumSharingCache]</a> .
<b>Per-node</b>	One instance per node. See 3.1.1 [Northbridge MSRs In Multi-Core Products].
<b>Not-same-for-all</b>	<p>Provide indication as to whether all instances of a given register should be the same across all cores/nodes according to the following equation:</p> $\text{SameOnAllCheckEnabled} = (\text{Writable} \&\& (\text{same-for-all} \mid \text{MSR}) \&\& \sim(\text{not-same-for-all} \parallel \text{UpdatedByHw})).$ $\text{UpdatedByHw} = (\text{Updated-by-hardware} \parallel \text{set-by-hardware} \parallel \text{cleared-by-hardware} \parallel \text{set-when-done} \parallel \text{cleared-when-done}).$
<b>Same-for-all</b>	
<b>Field Definitions</b>	
<b>Reserved</b>	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.



**Table 84: Terminology in Register Descriptions**

Term	Definition
<b>Unused</b>	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.
<b>MBZ</b>	Must be zero. If software attempts to set an MBZ bit to 1, a general-protection exception (#GP) occurs.
<b>RAZ</b>	Read as zero. Writes are ignored, unless RAZ is combined with write-only or write-1-only.
<b>Reset Definitions</b>	
<b>Reset</b>	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: <ul style="list-style-type: none"> <li>• <b>X</b>: an X in the reset value indicates that the field resets (warm or cold) to an unspecified state.</li> </ul>
<b>Cold reset</b>	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.
<b>Value</b>	The current value of a read-only field or register. A value statement explicitly defines the field or register as read-only and 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 on the node in a multi-core processor (e.g. MSRC001\_001F). 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. Some MSR's are conditionally shared; see D18F3x44[NbMcaToMstCpuEn].

### 3.1.2 Software Recommendation (BIOS, SBIOS)

The following keywords specify the recommended value to be set by software.

- BIOS: AMD BIOS.
- SBIOS: Platform BIOS.

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.

### 3.1.3 See Keyword (See:)

There is a special meaning applied to the use of “See:” that differs from the use of See not followed by a “:”.

- See, not followed by a “:”, simply refers the reader to a document location that contains related information.
- See followed by a “:” is a shorthand notation that indicates that the definition for this register or register field inherits all properties and definitions from the register or register field that follows "See:". Any def-

inition local to the register or register field supercedes this inheritance.

“See:” can be used in the following ways:

- Full register width. [CPUID Fn0000\\_0001\\_EAX](#) inherits it’s full register width definition from [D18F3xFC](#).
- Register field. [MSR0000\\_0277\[PA1MemType\]](#) inherits it’s definition from PA0MemType, however, the local reset of 4h overrides the inherited PA0MemType reset of 6h.
- Valid values definition. [MSR0000\\_020\[E,C,A,8,6,4,2,0\]\[MemType\]](#), for example, inherits the valid values definition from [Table 192 \[Valid Values for Memory Type Definition\]](#).

### 3.1.4 Mapping Tables

The following mapping table types are defined.

#### 3.1.4.1 Register Mapping

The register mapping table specifies the specific function for each register in a range of registers.

[Table 177](#), for example, specifies that the D18F5x160 function is for NB P-state 0.

#### 3.1.4.2 Index Mapping

The index mapping table is similar to the register mapping table, but specifies the register by index instead of by full register mnemonic.

[Table 136 \[Index Mapping for D18F2x9C\\_x0D0F\\_0\[F,8:0\]02\\_dct\[0\]\]](#), for example, specifies that the [D18F2x98\\_dct\[0\]\[31:0\]==0D0F\\_0002h](#), or [D18F2x9C\\_x0D0F\\_0002\\_dct\[0\]](#), function is for Byte 0.

#### 3.1.4.3 Field Mapping

The field mapping table maps the fields of a range of registers. The rows are the registers that are mapped. Each column specifies a field bit range that is mapped by that column for all registers. The cell at the intersection of the register and the field bit range specifies the suffix that is appended to the register field. “Reserved” specifies that the field is reserved for the register of that row.

#### 3.1.4.4 Broadcast Mapping

The broadcast mapping table maps a register address to a range of register addresses that are written as a group when the broadcast register address is written. The register address is formed by the concatenation of the row address with the column address. The cell at the intersection of the row and column address is a range of register addresses that will be written as a group when the row and column address is written.

#### 3.1.4.5 Reset Mapping

The reset mapping table specifies the reset, cold reset, or value for each register in a range of registers.

[Table 187 \[Reset Mapping for CPUID Fn8000\\_0000\\_E\[D,C,B\]X\]](#), for example, specifies that the [CPUID Fn0000\\_0000\\_EBX](#) register has a value of [6874\\_7541h](#), with a comment of “The ASCII characters “h t u A””.

#### 3.1.4.6 Valid Values

The valid values table defines the valid values for one or more register fields. The valid values table is equiva-

lent in function to the Bits/Description tables in register fields (E.g. [MSR0000\\_0277\[PA0MemType\]](#)) and is most often used when the table becomes too large and unwieldy to be included into the register field. (E.g. [Table 192 \[Valid Values for Memory Type Definition\]](#))

### 3.1.4.7 BIOS Recommendations

The BIOS recommendations table defines “BIOS:” recommendations that are conditional and complex enough to warrant a table.

[Table 165 \[BIOS Recommendations for D18F2x1B\[4:0\]\]](#), for example, specifies the BIOS recommendations for [D18F2x1B0\[DcqBwThrotWm\]](#) and [D18F2x1B4\[DcqBwThrotWm1, DcqBwThrotWm2\]](#). All cells under the “Condition” header for a given row are ANDed to form the condition for the values to the right of the condition. For example, rows 1-3 and column 1 provide the following equivalent BIOS recommendation:

- [D18F2x1B0\[DcqBwThrotWm\]](#): BIOS: IF ([DdrRate](#)==667) THEN 4h ELSEIF ([DdrRate](#)==800) THEN 5h ELSEIF ([DdrRate](#)==1066) THEN 6h ELSEIF etc.

## 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

Reset: 0. [IOCF8 \[IO-Space Configuration Address\]](#), and [IOCFC \[IO-Space Configuration Data Port\]](#), 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 IO link. Accesses to [IOCF8](#) and [IOCFC](#) received from an IO link are treated as all other IO transactions received from an IO link and are forwarded based on the settings in [D18F1x\[DC:C0\] \[IO-Space Base/Limit\]](#). [IOCF8](#) and [IOCFC](#) in the processor are not accessible from an IO link.

Bits	Description
31	<b>ConfigEn: configuration space enable.</b> Read-write. 1=IO read and write accesses to <a href="#">IOCFC</a> are translated into configuration cycles at the configuration address specified by this register. 0=IO read and write accesses are passed to the appropriate IO link and no configuration access is generated.
30:28	Reserved.
27:24	<b>ExtRegNo: extended register number.</b> Read-write. <a href="#">ExtRegNo</a> provides bits[11:8] and <a href="#">RegNo</a> provides bits[7:2] of the byte address of the configuration register. <a href="#">ExtRegNo</a> is reserved unless it is enabled by <a href="#">MSRC001_001F[EnableCf8ExtCfg]</a> .
23:16	<b>BusNo: bus number.</b> Read-write. Specifies the bus number of the configuration cycle.
15:11	<b>Device: bus number.</b> Read-write. Specifies the device number of the configuration cycle.
10:8	<b>Function.</b> Read-write. Specifies the function number of the configuration cycle.
7:2	<b>RegNo: register address.</b> Read-write. See <a href="#">IOCF8[ExtRegNo]</a> .
1:0	Reserved.

**IOFC IO-Space Configuration Data Port**

---

Bits	Description
31:0	<b>Data.</b> Read-write. Reset: 0. See <a href="#">IOCF8</a> .

### 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

Bits	Description
31:16	<b>DeviceID: device ID.</b> Read-only. Value: 1536h.
15:0	<b>VendorID: vendor ID.</b> Read-only. Value: 1022h.

#### D0F0x04 Status/Command

Reset: 0000\_0004h.

Bits	Description
31:21	Reserved.
20	<b>CapList: capability list.</b> Read-only. 1=Capability list supported.
19:3	Reserved.
2	<b>BusMasterEn: bus master enable.</b> Read-only.
1	<b>MemAccessEn: memory access enable.</b> Read-only.
0	<b>IoAccessEn: IO access enable.</b> Read-only.

#### D0F0x08 Class Code/Revision ID

Reset: 0600\_0000h.

Bits	Description
31:8	<b>ClassCode: class code.</b> Read-only. Provides the host bridge class code as defined in the PCI specification.
7:0	<b>RevID: revision ID.</b> Read-only.

#### D0F0x0C Header Type

Reset: 0080\_0000h.

Bits	Description
31:24	Reserved.
23	<b>DeviceType.</b> Read-only. 0b=Single function device. 1b=Multi-function device.
22:16	<b>HeaderType.</b> Read-only.
15:8	<b>LatencyTimer.</b> Read-only.
7:0	<b>CacheLineSize.</b> Read-only.

**D0F0x2C Subsystem and Subvendor ID**

Bits	Description
31:16	<b>SubsystemID</b> . Read-only. Value: 1410h.
15:0	<b>SubsystemVendorID</b> . Read-only. Value: 1022h.

**D0F0x34 Capabilities Pointer**

Reset: 0000\_0000h.

Bits	Description
31:8	Reserved.
7:0	<b>CapPtr: capabilities pointer</b> . Read-only. There is no capability list.

**D0F0x48 NB Header Write Register**

Reset: 0000\_0080h.

Bits	Description
31:8	Reserved.
7	<b>DeviceType: device type</b> . Read-write. This field sets the value in the corresponding field in <a href="#">D0F0x0C[DeviceType]</a> . 0b=Single function device. 1b=Multi-function device.
6:0	Reserved.

**D0F0x4C PCI Control**

Reset: 0000\_0000h.

Bits	Description
31:27	Reserved.
26	<b>HPDis: hot plug message disable</b> . Read-write. 1=Hot plug message generation is disabled.
25:24	Reserved.
23	<b>MMIOEnable: memory mapped IO enable</b> . Read-write. 1=Decoding of MMIO cycles is enabled. The MMIO Base/Limit pair ( <a href="#">D0F0x64_x17</a> and <a href="#">D0F0x64_x18</a> ) are decoded. This range is used to create an MMIO hole in the DRAM address range used for DMA decoding. DMA writes that fall into the MMIO range are treated as potential p2p requests. DMA reads that fall into the MMIO range are aborted as unsupported requests.
22:15	Reserved.
14:6	Reserved.
5	<b>SerrDis: system error message disable</b> . Read-write. 1=The generation of SERR messages is disabled.
4	<b>PMEDis: PME disable</b> . Read-write. 1=The generation of PME messages is disabled.
3:0	Reserved.

**D0F0x60 Miscellaneous Index**

Reset: 0000\_0000h. The index/data pair registers, **D0F0x60** and **D0F0x64**, are used to access the registers at **D0F0x64\_x[FF:00]**. To access any of these registers, the address is first written into the index register, **D0F0x60**, and then the data is read from or written to the data register, **D0F0x64**.

Bits	Description
31:7	Reserved.
6:0	<b>MiscIndAddr: miscellaneous index register address.</b> Read-write.

**D0F0x64 Miscellaneous Index Data**

See **D0F0x60**. Address: **D0F0x60[MiscIndAddr]**.

Bits	Description
31:0	<b>MiscIndData: miscellaneous index data register.</b>

**D0F0x64\_x00 Northbridge Control**

Reset: 0000\_0000h.

Bits	Description
31:8	Reserved.
7	<b>HwInitWrLock.</b> Read-write. 1=Lock HWInit registers. 0=Unlock HWInit registers.
6:0	Reserved.

**D0F0x64\_x0C IOC Bridge Control**

Reset: 0000\_0000h.

Bits	Description
31:0	Reserved.

**D0F0x64\_x0D IOC PCI Configuration**

Bits	Description
31:1	Reserved.
0	<b>PciDev0Fn2RegEn.</b> Read-write. Reset: 0. 1=Enable configuration accesses to device 0 function 2.

**D0F0x64\_x16 IOC Advanced Error Reporting Control**

Reset: 0000\_0001h.

Bits	Description
31:1	Reserved.
0	<b>AerUrMsgEn: AER unsupported request message enable.</b> Read-write. BIOS: 0. 1=AER unsupported request messages are enabled.

**D0F0x64\_x17 Memory Mapped IO Base Address**

Reset: 0000\_0000h.

Bits	Description
31:0	<b>MmioBase[47:16]: memory mapped IO base address.</b> Read-write.

**D0F0x64\_x18 Memory Mapped IO Limit**

Reset: 0000\_0000h.

Bits	Description
31:0	<b>MmioLimit[47:16]: memory mapped IO limit.</b> Read-write.

**D0F0x64\_x19 Top of Memory 2 Low**

Reset: 0000\_0000h.

Bits	Description
31:23	<b>Tom2[31:23]: top of memory 2.</b> Read-write. BIOS: <a href="#">MSRC001_001D</a> [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 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 or equal to this system address are master aborted.
22:1	Reserved.
0	<b>TomEn: top of memory enable.</b> Read-write. BIOS: <a href="#">MSRC001_0010</a> [MtrrTom2En]. 1=Top of memory check enabled.

**D0F0x64\_x1A Top of Memory 2 High**

Reset: 0000\_0000h.

Bits	Description
31:8	Reserved.
7:0	<b>Tom2[39:32]: top of memory 2.</b> Read-write. BIOS: <a href="#">MSRC001_001D</a> [TOM2[39:32]]. See <a href="#">D0F0x64_x19</a> [Tom2].



**D0F0x64\_x1D Internal Graphics PCI Control**

Reset: 0000\_0000h.

Bits	Description
31:4	Reserved.
3	<b>Vga16En: VGA IO 16 bit decoding enable.</b> Read-write. BIOS: <a href="#">D0F0x64_x1D[VgaEn]</a> . 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	<b>VgaEn: VGA enable.</b> Read-write. 1=Enable VGA range in Intgfx.
0	Reserved.

**D0F0x64\_x1F FCH Location**

Reset: 0002\_0001h.

Bits	Description																						
31:16	<b>SBLocatedCore: Indicates which GPP Core has the FCH attached to it.</b> Read-write. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>0000h</td> <td>No FCH attached.</td> </tr> <tr> <td>0001h</td> <td>FCH located under PPD.</td> </tr> <tr> <td>0002h</td> <td>FCH located under SBG.</td> </tr> <tr> <td>FFFFh-0003h</td> <td>Reserved.</td> </tr> </tbody> </table>	Bits	Definition	0000h	No FCH attached.	0001h	FCH located under PPD.	0002h	FCH located under SBG.	FFFFh-0003h	Reserved.												
Bits	Definition																						
0000h	No FCH attached.																						
0001h	FCH located under PPD.																						
0002h	FCH located under SBG.																						
FFFFh-0003h	Reserved.																						
15:0	<b>SBLocatedPort: Indicates which Port on the SBLocatedCore has the FCH.</b> Read-write. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>0000h</td> <td>No FCH attached.</td> </tr> <tr> <td>0001h</td> <td>FCH located on Port A of SBLocatedCore.</td> </tr> <tr> <td>0002h</td> <td>FCH located on Port B of SBLocatedCore.</td> </tr> <tr> <td>0003h</td> <td>Reserved.</td> </tr> <tr> <td>0004h</td> <td>FCH located on Port C of SBLocatedCore.</td> </tr> <tr> <td>0007h-0005h</td> <td>Reserved.</td> </tr> <tr> <td>0008h</td> <td>FCH located on Port D of SBLocatedCore.</td> </tr> <tr> <td>000Fh-0009h</td> <td>Reserved.</td> </tr> <tr> <td>0010h</td> <td>FCH located on Port E of SBLocatedCore.</td> </tr> <tr> <td>FFFFh-0011h</td> <td>Reserved.</td> </tr> </tbody> </table>	Bits	Definition	0000h	No FCH attached.	0001h	FCH located on Port A of SBLocatedCore.	0002h	FCH located on Port B of SBLocatedCore.	0003h	Reserved.	0004h	FCH located on Port C of SBLocatedCore.	0007h-0005h	Reserved.	0008h	FCH located on Port D of SBLocatedCore.	000Fh-0009h	Reserved.	0010h	FCH located on Port E of SBLocatedCore.	FFFFh-0011h	Reserved.
Bits	Definition																						
0000h	No FCH attached.																						
0001h	FCH located on Port A of SBLocatedCore.																						
0002h	FCH located on Port B of SBLocatedCore.																						
0003h	Reserved.																						
0004h	FCH located on Port C of SBLocatedCore.																						
0007h-0005h	Reserved.																						
0008h	FCH located on Port D of SBLocatedCore.																						
000Fh-0009h	Reserved.																						
0010h	FCH located on Port E of SBLocatedCore.																						
FFFFh-0011h	Reserved.																						

**D0F0x64\_x22 LCLK Control 0**

Reset: 7F3F\_8100h.

Bits	Description
31	Reserved.
30	<b>SoftOverrideClk0.</b> Read-write. BIOS: 0. 1=Dynamic clock gating disabled for the host request path to the PCIe® cores.
29	<b>SoftOverrideClk1.</b> Read-write. BIOS: 0. 1=Dynamic clock gating disabled for the host request path to the internal graphics and the host response path.

28	<b>SoftOverrideClk2.</b> Read-write. BIOS: 0. 1=Dynamic clock gating disabled for the host configuration requests.
27	<b>SoftOverrideClk3.</b> Read-write. BIOS: 0. 1=Dynamic clock gating disabled for the debug bus path.
26	<b>SoftOverrideClk4.</b> Read-write. BIOS: 0. 1=Dynamic clock gating disabled for the host request path to the configuration block.
25:0	Reserved.

### D0F0x64\_x23 LCLK Control 1

Reset: 7F3F\_8100h.

Bits	Description
31	Reserved.
30	<b>SoftOverrideClk0.</b> Read-write. BIOS: 0. 1=Dynamic clock gating disabled for upstream DMA requests from all sources.
29	<b>SoftOverrideClk1.</b> Read-write. BIOS: 0. 1=Dynamic clock gating disabled for upstream DMA requests from the GPPFCH link core.
28	<b>SoftOverrideClk2.</b> Read-write. BIOS: 0. 1=Dynamic clock gating disabled for upstream DMA requests from internal graphics and its DMA response reordering path.
27	<b>SoftOverrideClk3.</b> Read-write. BIOS: 0. 1=Dynamic clock gating disabled for upstream DMA requests from internal graphics.
26	<b>SoftOverrideClk4.</b> Read-write. BIOS: 0. 1=Dynamic clock gating disabled for upstream DMA requests from the Gfx link core.
25:0	Reserved.

### D0F0x64\_x3[4:0] Programmable Device Remap Register

Table 85: Reset values for D0F0x64\_x3[4:0]

Register	Reset	Function
D0F0x64_x30	0000_0011h	Program [7:3]DevNum, [2:0]FnNum to map to PortA of PPD.
D0F0x64_x31	0000_0012h	Program [7:3]DevNum, [2:0]FnNum to map to PortB of PPD.
D0F0x64_x32	0000_0013h	Program [7:3]DevNum, [2:0]FnNum to map to PortC of PPD.
D0F0x64_x33	0000_0014h	Program [7:3]DevNum, [2:0]FnNum to map to PortD of PPD.
D0F0x64_x34	0000_0015h	Program [7:3]DevNum, [2:0]FnNum to map to PortE of PPD.

Software can only utilize device and function number combinations that are used by other (local) PCIe<sup>®</sup> bridges. This effectively allows swapping of device and function numbers between bridges.

Bits	Description
31:8	Reserved.
7:0	<b>DevFnMap.</b> Read-write. Program [7:3]DevNum, [2:0]FnNum to map to PortA/B/C/D of each PCIe <sup>®</sup> core.

**D0F0x64\_x46 IOC Features Control**

Reset: 0001\_1063h.

Bits	Description										
31:28	Reserved.										
27:24	Reserved.										
23	Reserved.										
22	Reserved.										
21:17	Reserved.										
16	<b>CggtLclkOverride.</b> Read-write. BIOS: 0. Global bit to disable all LCLK gating branches.										
15:5	Reserved.										
4:3	Reserved.										
2:1	<b>P2PMode: peer-to-peer mode.</b> Read-write. Specifies how upstream write transactions above <a href="#">D0F0x64_x19</a> [Tom2] are completed. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Mode 0. 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.</td> </tr> <tr> <td>01b</td> <td>Mode 1. Forward writes to the host bridge that do not hit one of the internal PCI bridges. Forward writes that hit one of the internal PCI bridges to the bridge.</td> </tr> <tr> <td>10b</td> <td>Mode 2. Forward all writes to the host bridge0.</td> </tr> <tr> <td>11b</td> <td>Reserved.</td> </tr> </tbody> </table>	Bits	Definition	00b	Mode 0. 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.	01b	Mode 1. Forward writes to the host bridge that do not hit one of the internal PCI bridges. Forward writes that hit one of the internal PCI bridges to the bridge.	10b	Mode 2. Forward all writes to the host bridge0.	11b	Reserved.
Bits	Definition										
00b	Mode 0. 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.										
01b	Mode 1. Forward writes to the host bridge that do not hit one of the internal PCI bridges. Forward writes that hit one of the internal PCI bridges to the bridge.										
10b	Mode 2. Forward all writes to the host bridge0.										
11b	Reserved.										
0	Reserved.										

**D0F0x7C IOC Configuration Control**

Cold reset: 0000\_0000h.

Bits	Description
31:1	Reserved.
0	<b>ForceIntGfxDisable: internal graphics disable.</b> Read-write. Setting this bit disables the internal graphics and the HD Audio controller.

**D0F0x84 Link Arbitration**

Bits	Description
31:10	Reserved.
9	<b>PmeTurnOff: PME_Turn_Off message trigger.</b> Read-write. Reset: 0. 1=Trigger a PME_Turn_Off message to all downstream devices if PmeMode=1.
8	<b>PmeMode: PME message mode.</b> Read-write. Reset: 0. 1=PME_Turn_Off message is triggered by writing PmeTurnOff. 0=PME_Turn_Off message is triggered by a message from the FCH.
7:4	Reserved.

3	<b>VgaHole: vga memory hole.</b> Read-write. Reset: 1. This bit creates a hole in memory for the VGA memory range. 1=Requests hitting the VGA range are checked against PCI bridge memory ranges instead of being forwarded to system memory.
2:0	Reserved.

### **D0F0x90 Northbridge Top of Memory**

Reset: 0000\_0000h.

Bits	Description
31:23	<b>TopOfDram.</b> Read-write. BIOS: <a href="#">MSRC001_001A</a> [TOM[31:23]]. Specifies the address that divides between MMIO and DRAM. From TopOfDram to 4G is MMIO; below TopOfDram is DRAM.
22:0	Reserved.

### **D0F0x94 Northbridge ORB Configuration Offset**

Reset: 0000\_0000h.

The index/data pair registers, [D0F0x94](#) and [D0F0x98](#), are used to access the registers at [D0F0x98\\_x](#)[FF:00]. To access any of these registers, the address is first written into the index register, [D0F0x94](#), and then the data is read from or written to the data register, [D0F0x98](#).

Bits	Description
31:7	Reserved.
6:0	<b>OrbIndAddr: ORB index register address.</b> Read-write.

### **D0F0x98 Northbridge ORB Configuration Data Port**

See [D0F0x94](#). Address: [D0F0x94](#)[OrbIndAddr].

Bits	Description
31:0	<b>OrbIndData: ORB index data register.</b>

### **D0F0x98\_x06 ORB Downstream Control 0**

Reset: 0000\_0000h.

Bits	Description
31:27	Reserved.
26	<b>UmiNpMemWrEn.</b> Read-write. BIOS: See <a href="#">2.11.4</a> . 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\_0080h.

Bits	Description						
31	<b>SMUCsrIsocEn.</b> Read-write. BIOS: 1. 1=CSR accesses go through ISOC channel. If this bit is set, <a href="#">D0F0x98_x1E[HiPriEn]</a> must also be set.						
30:17	Reserved.						
16	<b>SyncFloodOnParityErr.</b> Read-write. Enable short circuit syncflood when arb_np detects a parity error for error containment.						
15	<b>DropZeroMaskWrEn.</b> Read-write. BIOS: 1. 1=Drop byte write request that have all bytes masked. 0=Forward byte write request that have all bytes masked.						
14:8	Reserved.						
7	<b>IommuIsocPassPWMode.</b> Read-write. BIOS: 1. 1=Always set PassPW for IOMMU upstream isochronous requests.						
6	<b>DmaReqRespPassPWMode.</b> Read-write. BIOS: 0. Specifies the RespPassPW bit for non-posted upstreamDMA requests. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Bit</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Always 1.</td> </tr> <tr> <td>1</td> <td>Value passed from IOC.</td> </tr> </tbody> </table>	Bit	Description	0	Always 1.	1	Value passed from IOC.
Bit	Description						
0	Always 1.						
1	Value passed from IOC.						
5	Reserved.						
4	<b>IommuBwOptEn.</b> Read-write. BIOS: 1. 1=Optimize IOMMU L2 byte write by detecting consecutive DW mask and translate the request to DW write.						
3	Reserved.						
2	<b>IocRdROMapDis.</b> Read-write. 1=Disable mapping relax ordering bit to RdRespPpw bit for IOC reads.						
1	<b>IocWrROMapDis.</b> Read-write. 1=Disables mapping relax ordering bit to PassPw bit for IOC writes.						
0	<b>IocBwOptEn.</b> Read-write. BIOS: 1. 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**

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	<b>NpWrrLenD.</b> Read-write. Reset: 8h. BIOS: 8h. This field defines the maximum number of non-posted read requests from the SPG that are serviced before the arbiter switches to the next client.
23:16	<b>NpWrrLenC.</b> Read-write. Reset: 8h. BIOS: 1h. This field defines the maximum number of non-posted read requests from the SMU that are serviced before the arbiter switches to the next client.

15:8	<b>NpWrrLenB.</b> Read-write. Reset: 8h. BIOS: 8h. This field defines the maximum number of non-posted read requests from IOMMU that are serviced before the arbiter switches to the next client.
7:0	<b>NpWrrLenA.</b> Read-write. Reset: 8h. BIOS: 8h. 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: 0000\_0808h.

This register specifies the weights of the weighted round-robin arbiter in stage 1 of the upstream arbitration for posted writes.

Bits	Description
31:16	Reserved.
15:8	<b>PWrrLenB.</b> Read-write. This field defines the maximum number of posted write requests from the IOMMU that are serviced before the arbiter switches to the next client.
7:0	<b>PWrrLenA.</b> 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
31:24	Reserved.
23:16	Reserved.
15:8	<b>GcmWrrLenB.</b> 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.
7:0	<b>GcmWrrLenA.</b> 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.

### **D0F0x98\_x1E ORB Receive Control 0**

Reset: 4800\_0000h.

Bits	Description
31:24	<b>RxErrStatusDelay.</b> Read-write. BIOS: 48h. Delay error status by number of LCLK cycles to filter false errors caused by reset skew.
23:2	Reserved.

1	<b>HiPriEn.</b> Read-write. BIOS: 1. 1=High priority channel enabled. See <a href="#">D0F0x98_x27[UrRedirectAddr[31:6]]</a> . IF ( <a href="#">D0F0x98_x1E[HiPriEn]</a> ==0) THEN ( <a href="#">D0F0x98_x07[SMUCsrIsocEn]</a> ==0). IF ( <a href="#">D0F0x98_x1E[HiPriEn]</a> ==1) THEN ( <a href="#">D18F0x[E4,C4,A4,84][IsocEn]</a> ==1) in order to fully enable the Isoc channel on the ONION Link.
0	Reserved.

**D0F0x98\_x26 ORB IOMMU Control 0**

Reset: 0000\_0000h.

Bits	Description
31:8	Reserved.
7:0	<b>UrRedirectAddr[39:32].</b> Read-write. See: <a href="#">D0F0x98_x27[UrRedirectAddr[31:6]]</a> .

**D0F0x98\_x27 ORB IOMMU Control 1**

Reset: 0000\_0000h.

Bits	Description
31:6	<b>UrRedirectAddr[31:6].</b> Read-write. BIOS: UrRedirectAddr[39:6] must be programmed to a safe system memory address. UrRedirectAddr[39:6] = { <a href="#">D0F0x98_x26[UrRedirectAddr[39:32]]</a> , UrRedirectAddr[31:6]}. IOMMU requests that are not directed to system memory are redirected to UrRedirectAddr.
5:0	Reserved.

**D0F0x98\_x28 ORB Transmit Control 0**

Reset: 0000\_0002h.

Bits	Description
31:2	Reserved.
1	<b>ForceCoherentIntr.</b> Read-write. BIOS: 1. 1=Interrupt request are forced to have coherent bit set.
0	Reserved.

**D0F0x98\_x2C ORB Clock Control**

Reset: 000F\_0204h.

Bits	Description
31:16	<b>WakeHysteresis.</b> Read-write. BIOS: 19h. Specifies the amount of time hardware waits after ORB becomes idle before deasserting the wake signal to the NB. Wait time = WakeHysteresis * 200ns. Changes to this field should be done prior to setting DynWakeEn.
15:10	Reserved.
9	<b>SBDmaActiveMask.</b> Read-write. BIOS: 1. 0=SB_DMA_ACTIVE_L state affects OnInbWake state. 1= SB_DMA_ACTIVE_L state is masked out.

8:3	Reserved.
2	<b>CgctlLclkOverride</b> . Read-write. BIOS: 0h. Global bit to disable all LCLK gating branches in the ORB.
1	<b>DynWakeEn</b> . Read-write. BIOS: 1. 1=Enable dynamic toggling of the wake signal between ORB and NB. 0=Disable dynamic toggling of the wake signal. See WakeHysteresis.
0	Reserved.

### D0F0x98\_x37 ORB Allow LDTSTOP Control 0

Reset: 0020\_0000h.

Bits	Description
31:28	Reserved.
27:16	<b>LDTStopHystersis</b> . Read-write. Specifies the number of timer periods (200 ns) the AllowLDTStop signal is held low before ORB asserts the signal again.
15:2	Reserved.
1	<b>DmaActiveOutEn</b> . Read-write. 1=Enable ORB to drive the DMAACTIVE_L pin. Meaningful only when <code>D0F0x98_x37[AllowLDTStopPinMode]=0</code> .
0	<b>AllowLDTStopPinMode</b> . Read-write. Indicates the definition of the ALLOW_LDTSTOP pin. 0=Pin is used as DMAACTIVE_L. 1=Pin is used as ALLOW_LDTSTOP.

### D0F0x98\_x3A ORB Source Tag Translation Control 2

Reset: 0000\_0000h.

Bits	Description
31:0	<b>ClumpingEn</b> . Read-write. BIOS should follow the below requirements. Valid only for PPD and GBIF client clumping; internal unit ID ranges 4h-8h and 14h-17h respectively. Legal PPD clumping settings are: [8:4]=00010b, applicable only in x0/0/0/0/8 system configuration. Legal GBIF clumping settings are: [23:20]=0010b, 0110b and 1110b which are applicable in any system configuration. 1110b is the recommended value. All other bits of this register must always remain 0. See <a href="#">D18F0x[11C,118,114,110]</a> .

### D0F0x98\_x3B ORB Source Tag Translation Control 3

Reset: 0000\_0000h.

Bits	Description
31:0	<b>IocOutstandingMask</b> . Read-write. Limit number of outstanding requests for very dma client via the IOC.



**D0F0x98\_x4[A,9] ORB LCLK Clock Control 1-0**

Reset: 7F3F\_8100h.

Bits	Description
31	Reserved.
30	<b>SoftOverrideClk0.</b> Read-write. BIOS: 0. See SoftOverrideClk6.
29	<b>SoftOverrideClk1.</b> Read-write. BIOS: 0. See SoftOverrideClk6.
28	<b>SoftOverrideClk2.</b> Read-write. BIOS: 0. See SoftOverrideClk6.
27	<b>SoftOverrideClk3.</b> Read-write. BIOS: 0. See SoftOverrideClk6.
26	<b>SoftOverrideClk4.</b> Read-write. BIOS: 0. See SoftOverrideClk6.
25	<b>SoftOverrideClk5.</b> Read-write. BIOS: 0. See SoftOverrideClk6.
24	<b>SoftOverrideClk6.</b> Read-write. BIOS: 0. 1=Clock gating disabled. 0=Clock gating enabled.
23:0	Reserved.

**D0F0xB8 SMU Index Address**

The index/data pair registers, [D0F0xB8](#) and [D0F0xBC](#), are used to access the registers at [D0F0xBC\\_x\[FFFFFFFF:00000000\]](#). To access any of these registers, the address is first written into the index register, [D0F0xB8](#), and then the data is read from or written to the data register, [D0F0xBC](#).

Bits	Description
31:0	<b>NbSmuIndAddr: smu index address.</b> Read-write. Reset: 0.

**D0F0xBC SMU Index Data**See [D0F0xB8](#). Address: [D0F0xB8\[NbSmuIndAddr\]](#).

Bits	Description
31:0	<b>NbSmuIndData: smu index data.</b> Read-write

**D0F0xBC\_x3F800 FIRMWARE\_FLAGS**

Reset: xxxx\_xxxxh.

Bits	Description						
31:24	<b>TestCount.</b> Read-write. Test count.						
23:1	Reserved.						
0	<b>InterruptsEnabled.</b> Read-write. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Firmware has not yet enabled interrupts. BIOS/Driver cannot yet send message interrupts to SMC.</td> </tr> <tr> <td>1</td> <td>Firmware has enabled interrupts. BIOS/Driver can send message interrupts to SMC.</td> </tr> </tbody> </table>	Bits	Definition	0	Firmware has not yet enabled interrupts. BIOS/Driver cannot yet send message interrupts to SMC.	1	Firmware has enabled interrupts. BIOS/Driver can send message interrupts to SMC.
Bits	Definition						
0	Firmware has not yet enabled interrupts. BIOS/Driver cannot yet send message interrupts to SMC.						
1	Firmware has enabled interrupts. BIOS/Driver can send message interrupts to SMC.						

**D0F0xBC\_x3F804 FIRMWARE\_VID**

Reset: xxxx\_xxxxh.

Bits	Description
7:0	<b>FirmwareVid.</b> Read-write. Current voltage set by firmware voltage controller.

**D0F0xBC\_x3F820 PM\_INTERVAL\_CNTL\_0**

Reset: xxxx\_xxxxh.

Bits	Description
31:24	<b>Loadline.</b> Read-write.
23:16	<b>VoltageCntl.</b> Read-write.
15:8	<b>ThermalCntl.</b> Read-write.
7:0	<b>LclkDpm.</b> Read-write.

**D0F0xBC\_x3F828 PM\_TIMER\_PERIOD**

Bits	Description
31:0	<b>TimerPeriod.</b> Read-write. Reset: X. Specifies the period at which various power management related algorithms are run. Period = TimerPeriod / REFCLK.

**D0F0xBC\_x3F9D8 PM\_CONFIG**

Reset: xxxx\_xxxxh.

Bits	Description
31:30	Reserved.

29	<b>SviMode.</b> Read-write. <u>Bits</u> <u>Description</u> 0                              SVI1. 1                              SVI2.
28	<b>BapmCoeffOverride.</b> Read-write.
27	<b>NbPstateAllCpusIdle.</b> Read-write. BIOS: For CPU P-state associated with <a href="#">D18F3xA8</a> [PopDownPstate], IF ( <a href="#">MSRC001_00</a> [6B:64][NbPstate]==0) THEN 1. ELSE 0. ENDIF. <u>Bits</u> <u>Description</u> 0                              Use low NB P-state voltage when AllCpusIdle. 1                              Use high NB P-state voltage when AllCpusIdle.
26:24	<b>PstateAllCpusIdle.</b> Read-write. BIOS: <a href="#">D18F3xA8</a> [PopDownPstate].
23:6	Reserved.
5	<b>EnableNbDpm.</b> Read-write. <u>Bits</u> <u>Description</u> 0                              Disable Dynamic NB Pstate Management. Should be cleared before sending message to SMU. 1                              Enable Dynamic NB Pstate Management. Should be set before sending BIOSMC_MSG_NBDPM_Enable message to SMU.
4	Reserved.
3	<b>EnableLpmx.</b> Read-write.
2	<b>EnableTdcLimit.</b> Read-write.
1	<b>EnableBapm.</b> Read-write.
0	<b>EnableVpcAccumulators.</b> Read-write.

**D0F0xBC\_x3F9E8 NB\_DPM\_CONFIG\_1**

Reset: xxxx\_xxxxh.

Bits	Description
31:24	<b>DpmXNbPsHi.</b> Read-write. See: <a href="#">Dpm0PgNbPsLo</a> .
23:16	<b>DpmXNbPsLo.</b> Read-write. See: <a href="#">Dpm0PgNbPsLo</a> .
15:8	<b>Dpm0PgNbPsHi.</b> Read-write. See: <a href="#">Dpm0PgNbPsLo</a> .
7:0	<b>Dpm0PgNbPsLo.</b> Read-write. Indexes the NB P-state used during specific levels of GPU activity. See <a href="#">2.5.4.1 [NB P-states]</a> . <u>Bits</u> <u>NB P-state Indexed</u> 00b                      D18F3x160 (see <a href="#">D18F5x16</a> [C:0]). 01b                      D18F3x164 (see <a href="#">D18F5x16</a> [C:0]). 10b                      D18F3x168 (see <a href="#">D18F5x16</a> [C:0]). 11b                      D18F3x16C (see <a href="#">D18F5x16</a> [C:0]).

**D0F0xBC\_x3F9EC NB\_DPM\_CONFIG\_2**

Reset: xxxx\_xxxxh.

Bits	Description
31:25	Reserved.
24	<b>EnableNbPsi1</b> . Read-write. Specifies how PSII_L functions for VDDNB. 0=PSII_L is deasserted. 1=PSII_L is asserted whenever the GPU is idle.
23:17	Reserved.
16	<b>SkipDPM0</b> . Read-write. Specifies whether SMU waits for SCLK DPM to transition to state 0 before transitioning NB to the NB P-states indexed Dpm0PgNbPsHi and Dpm0PgNbPsLo. 0=Wait for SCLK DPM state 0. 1=Do not wait for SCLK DPM state 0. See <a href="#">2.5.4.1 [NB P-states]</a> .
15:9	Reserved.
8	<b>SkipPG</b> . Read-write. Specifies whether SMU waits for the GPU to be power gated before transitioning NB to the NB P-states indexed Dpm0PgNbPsHi and Dpm0PgNbPsLo. 0=Wait for GPU power gating. 1=Do not wait for GPU power gating. See <a href="#">2.5.4.1 [NB P-states]</a> .
7:0	<b>Hysteresis</b> . Read-write. Specifies the time the GPU must be idle before transitioning to the NB P-states indexed by Dpm0PgNbPsHi and Dpm0PgNbPsLo.

**D0F0xBC\_x3F9F4 CSR\_GNB\_1**

Bits	Description
31:24	<b>SviTrimValueVddNB</b> . Read-write. Reset: x. BIOS: <a href="#">D18F5x188[NbLoadLineTrim]</a> .
23:16	<b>SviTrimValueVdd</b> . Read-write. Reset: x. BIOS: <a href="#">D18F5x12C[CoreLoadLineTrim]</a> .
15:0	Reserved.

**D0F0xBC\_x3F9F8 CSR\_GNB\_3**

Reset: xxxx\_xxxxh.

Bits	Description
31:16	Reserved.
15:8	<b>SviLoadLineOffsetVddNB</b> . Read-write. Reset: x. BIOS: <a href="#">D18F5x188[NbOffsetTrim]</a> .
7:0	<b>SviLoadLineOffsetVdd</b> . Read-write. Reset: x. BIOS: <a href="#">D18F5x12C[CoreOffsetTrim]</a> .

**D0F0xBC\_x3FD[8C:00:step14] LCLK DPM Control 0**

Reset: xxxx\_xxxxh. See [2.5.6.1.3 \[LCLK DPM\]](#). Each register in [D0F0xBC\\_x3FD\[8C:00:step14\]](#) corresponds to one LCLK DPM state as follows.

Table 86: Register Mapping for [D0F0xBC\\_x3FD\[8C:00:step14\]](#)

Register	Function	Register	Function
D0F0xBC_x3FD00	State 0	D0F0xBC_x3FD50	State 4

Table 86: Register Mapping for D0F0xBC\_x3FD[8C:00:step14]

D0F0xBC_x3FD14	State 1	D0F0xBC_x3FD64	State 5
D0F0xBC_x3FD28	State 2	D0F0xBC_x3FD78	State 6
D0F0xBC_x3FD3C	State 3	D0F0xBC_x3FD8C	State 7

Bits	Description
31:24	<b>StateValid.</b> Read-write. 1=DPM state is valid. 0=DPM state is invalid.
23:16	<b>LclkDivider.</b> Read-write. Specifies the LCLK divisor for this DPM state.
15:8	<b>VID.</b> Read-write. Specifies the VDDNB VID for this DPM state.
7:0	<b>LowVoltageReqThreshold.</b> Read-write.

**D0F0xBC\_x3FD[94:08:step14] LCLK DPM Control 2**

Reset: xxxx\_xxxxh. Each register in D0F0xBC\_x3FD[94:08:step14] corresponds to one LCLK DPM state as follows.

Table 87: Register Mapping for D0F0xBC\_x3FD[94:08:step14]

Register	Function	Register	Function
D0F0xBC_x3FD08	State 0	D0F0xBC_x3FD58	State 4
D0F0xBC_x3FD1C	State 1	D0F0xBC_x3FD6C	State 5
D0F0xBC_x3FD30	State 2	D0F0xBC_x3FD80	State 6
D0F0xBC_x3FD44	State 3	D0F0xBC_x3FD94	State 7

Bits	Description
31:16	<b>ResidencyCounter.</b> Read-write.
15:8	<b>HysteresisUp.</b> Read-write.
7:0	<b>HysteresisDown.</b> Read-write.

**D0F0xBC\_x3FD[9C:10:step14] LCLK DPM Activity Thresholds**

Reset: xxxx\_xxxxh. Each register in D0F0xBC\_x3FD[9C:10:step14] corresponds to one LCLK DPM state as follows.

Table 88: Register Mapping for D0F0xBC\_x3FD[9C:10:step14]

Register	Function	Register	Function
D0F0xBC_x3FD10	State 0	D0F0xBC_x3FD60	State 4
D0F0xBC_x3FD24	State 1	D0F0xBC_x3FD74	State 5
D0F0xBC_x3FD38	State 2	D0F0xBC_x3FD88	State 6
D0F0xBC_x3FD4C	State 3	D0F0xBC_x3FD9C	State 7

Bits	Description
31:24	<b>ActivityThreshold.</b> Read-write.

23:16	EnabledForThrottle. Read-write.
15:0	Reserved.

**D0F0xBC\_x3FDC8 SMU\_LCLK\_DPM\_CNTL**

Reset: xxxx\_xxxxh.

Bits	Description
31:24	<b>LclkDpmEn.</b> Read-write. 1b=Enable LCLK DPM
23:16	<b>VoltageChgEn.</b> Read-write. 1=Enable voltage change during LCLK DPM state transition.
15:8	<b>LclkDpmBootState.</b> Read-write.
7:0	Reserved.

**D0F0xBC\_x3FDD0 SMU\_LCLK\_DPM\_THERMAL\_THROTTLING\_CNTL**

Reset: xxxx\_xxxxh.

Bits	Description
31:24	<b>TtHtcActive.</b> Read-write.
23:16	<b>LclkTtMode.</b> Read-write.
15:8	<b>TemperatureSel.</b> Read-write.
7:0	<b>LclkThermalThrottlingEn.</b> Read-write.

**D0F0xBC\_x3FDD4 SMU\_LCLK\_DPM\_THERMAL\_THROTTLING\_THRESHOLDS**

Reset: xxxx\_xxxxh.

Bits	Description
31:16	<b>HighThreshold.</b> Read-write. Specifies the high thermal threshold for LCLK thermal throttling.
15:0	<b>LowThreshold.</b> Read-write. Specifies the low thermal threshold for LCLK thermal throttling.

**D0F0xBC\_xC020\_008C LCLK\_DEEP\_SLEEP\_CNTL**

Bits	Description
------	-------------

31:3	Reserved.																		
2:0	<b>DivId.</b> Read-write. Reset: 5. BIOS: 100b.																		
	<table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>Clock OFF</td> </tr> <tr> <td>001b</td> <td>Divide by 2</td> </tr> <tr> <td>010b</td> <td>Divide by 4</td> </tr> <tr> <td>011b</td> <td>Divide by 8</td> </tr> <tr> <td>100b</td> <td>Divide by 16</td> </tr> <tr> <td>101b</td> <td>Divide by 32</td> </tr> <tr> <td>110b</td> <td>Reserved</td> </tr> <tr> <td>111b</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Description	000b	Clock OFF	001b	Divide by 2	010b	Divide by 4	011b	Divide by 8	100b	Divide by 16	101b	Divide by 32	110b	Reserved	111b	Reserved
Bits	Description																		
000b	Clock OFF																		
001b	Divide by 2																		
010b	Divide by 4																		
011b	Divide by 8																		
100b	Divide by 16																		
101b	Divide by 32																		
110b	Reserved																		
111b	Reserved																		

### **D0F0xBC\_xC020\_0110 Activity Monitor Control**

Bits	Description												
31:11	Reserved.												
10	<b>EnOrbDsCnt.</b> Read-write. Reset: X. 1=Enable downstream counter.												
9	<b>EnOrbUsCnt.</b> Read-write. Reset: X. 1=Enable upstream counter.												
8	<b>EnBifCnt.</b> Read-write. Reset: X. 1=Enable BIF counter.												
7:5	Reserved.												
4:3	<b>BusyCntSel.</b> Read-write. Reset: 0. Specifies subcomponents or activity monitored by the LCLK activity monitor.												
	<table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>GFX DMA (BIF)</td> <td>10b</td> <td>Downstream activity</td> </tr> <tr> <td>01b</td> <td>Upstream activity</td> <td>11b</td> <td>Up/downstream activity max</td> </tr> </tbody> </table>	Bits	Definition	Bits	Definition	00b	GFX DMA (BIF)	10b	Downstream activity	01b	Upstream activity	11b	Up/downstream activity max
Bits	Definition	Bits	Definition										
00b	GFX DMA (BIF)	10b	Downstream activity										
01b	Upstream activity	11b	Up/downstream activity max										
2	Reserved.												
1	<b>PeriodCntRst.</b> Read-write. Reset: X.												
0	<b>ActivityCntRst.</b> Read-write. Reset: X.												

### **D0F0xBC\_xC210\_0000 CPU Interrupt Request**

See 2.12.1 [Software Interrupts].

Bits	Description
31:17	Reserved.
16:1	<b>ServiceIndex.</b> Read-write. Reset: 0.
0	<b>IntToggle.</b> Read-write. Reset: 0.

### **D0F0xBC\_xC210\_0004 CPU Interrupt Status**

See 2.12.1 [Software Interrupts].

Bits	Description
31:2	Reserved.

1	<b>IntDone.</b> Read-only; updated-by-hardware. Reset: 0.
0	<b>IntAck.</b> Read-only; updated-by-hardware. Reset: 0.

### D0F0xBC\_xC210\_003C CPU Interrupt Argument

See 2.12.1 [Software Interrupts].

Bits	Description
31:0	<b>Argument.</b> Read-write. Reset: 0. Optional argument for a software interrupt.

### D0F0xBC\_xC210\_0040 CPU Interrupt Response

See 2.12.1 [Software Interrupts].

Bits	Description
31:0	<b>Argument.</b> Read-write. Reset: 0. Optional response data upon completing a software interrupt.

### D0F0xC8 DEV Index Address

The index/data pair registers, [D0F0xC8](#) and [D0F0xCC](#) are used to access the registers at [D0F0xCC\\_x\[FF:00\]](#). To access any of these registers, the address is first written into the index register, [D0F0xC8](#), and then the data is read from or written to the data register, [D0F0xCC](#). Specific bridges (Device/Function) are selected using the [D0F0xC8\[NbDevIndSel\]](#) field and enumerated as *\_ib[15:11]* in the indexed register's mnemonic.

Bits	Description																								
31:24	Reserved.																								
23:16	<b>NbDevIndSel: Device selector.</b> Read-write. Reset: 0. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>10h-00h</td> <td>Reserved</td> <td>15h</td> <td>D2F5</td> </tr> <tr> <td>11h</td> <td>D2F1</td> <td>FFh-16h</td> <td>Reserved</td> </tr> <tr> <td>12h</td> <td>D2F2</td> <td></td> <td></td> </tr> <tr> <td>13h</td> <td>D2F3</td> <td></td> <td></td> </tr> <tr> <td>14h</td> <td>D2F4</td> <td></td> <td></td> </tr> </tbody> </table>	Bits	Definition	Bits	Definition	10h-00h	Reserved	15h	D2F5	11h	D2F1	FFh-16h	Reserved	12h	D2F2			13h	D2F3			14h	D2F4		
Bits	Definition	Bits	Definition																						
10h-00h	Reserved	15h	D2F5																						
11h	D2F1	FFh-16h	Reserved																						
12h	D2F2																								
13h	D2F3																								
14h	D2F4																								
15:7	Reserved.																								
6:0	<b>NbDevIndAddr: Bridge (Device) index address.</b> Read-write. Reset: 0.																								

### D0F0xCC DEV Index Data

See [D0F0xC8](#). Address: [D0F0xC8\[NbDevIndAddr\]](#).

Bits	Description
31:0	<b>NbDevIndData: dev index data.</b>

### D0F0xCC\_x01 IOC Bridge Control

Reset: 0000\_0000h.



Bits	Description
31:24	<b>ApicRange</b> . Read-write. Sets the bridge APIC range.
23	<b>ApicEnable</b> . Read-write. 1=Enables the bridge APIC range decoding. Requests fall in bridge APIC range if addr[39:12]={20'h00_FEC, APIC_Range[7:0]}.
22:21	Reserved.
20	<b>SetPowEn</b> . Read-write. 1=Enable generation of set_slot_power message to the bridge.
19	Reserved.
18	<b>CrsEnable</b> . Read-write. 1=Enables the hardware retry on receiving configuration request retry status.
17	<b>ExtDevCrsEn</b> . Read-write. 1=Reset the bridge CRS counter when an external device is plugged in or the link is down.
16	<b>ExtDevPlug</b> . Read-write. 1=Indicates to IOC that an external device is being plugged on the bridge.
15:4	Reserved.
3	<b>P2pDis</b> . Read-write. 1=Disables local peer-to-peer transactions forwarded to this bridge.
2	<b>CfgDis</b> . Read-write. 1=Configuration accesses to this bridge are disabled. Non-FCH bridges are not expected to set this bit.
1	<b>BusMasterDis</b> . Read-write. 1=The bridge's ability to operate as a bus master is disabled. This overrides the Bus Master Enable bit in the bridge.
0	<b>BridgeDis</b> . Read-write. 1=The bridge is hidden and no accesses are allowed to this bridge.

#### **D0F0xD4\_x0109\_14C3 Bif Doorbell Control Ind**

Reset: 0000\_0000h.

Bits	Description
31:0	Reserved.

#### **D0F0xD4\_x0109\_14E1 CC Bif Bx Strap0 Ind**

Reset: 0000\_C004h.

Bits	Description
31:13	Reserved.
12	<b>StrapBifDoorbellBarDis</b> . Read-write.
11:6	Reserved.
5:3	<b>StrapBifMemApSize</b> . Read-write.
2:1	<b>StrapBifRegApSize</b> . Read-write.
0	Reserved.

#### **D0F0xD4\_x0109\_14E2 CC Bif Bx Strap1 Ind**

Reset: 0000\_0000h.

Bits	Description
31:14	Reserved.
13	Reserved.
12:11	Reserved.
10	Reserved.
9	Reserved.
8	Reserved.
7:4	Reserved.
3	<b>StrapBifF064BarDisA.</b> Read-write.
2	Reserved.
1	<b>StrapBifIoBarDis.</b> Read-write.
0	Reserved.

#### **D0F0xD4\_x0109\_1507 CC Bif Bx Pinstrap0 Ind**

Reset: 0000\_0802h.

Bits	Description
31:17	Reserved.
16	Reserved.
15:8	Reserved.
7:5	<b>StrapBifMemApSizePin.</b> Read-write.
4:0	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\\_x0\[2:1\]xxx\\_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\[31:0\]=0120\\_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\[31:0\]=0120\\_5602h](#).

Bits	Description												
31:24	<b>BlockSelect: block select.</b> Read-write. This field is used to select the specific register block to access. The encodings supported depends on the FrameType selected. <table border="1"> <thead> <tr> <th>FrameType</th> <th>Encoding</th> </tr> </thead> <tbody> <tr> <td>01h</td> <td>1=GPP link core</td> </tr> <tr> <td>10h</td> <td>1=Phy interface 0</td> </tr> <tr> <td>20h</td> <td>1=Phy 0</td> </tr> <tr> <td>30h</td> <td>1=Wrapper</td> </tr> <tr> <td>40h</td> <td>1=IO link core</td> </tr> </tbody> </table>	FrameType	Encoding	01h	1=GPP link core	10h	1=Phy interface 0	20h	1=Phy 0	30h	1=Wrapper	40h	1=IO link core
FrameType	Encoding												
01h	1=GPP link core												
10h	1=Phy interface 0												
20h	1=Phy 0												
30h	1=Wrapper												
40h	1=IO link core												
23:16	<b>FrameType: frame type.</b> Read-write. This field is used to select the type of register block to access. <table border="1"> <thead> <tr> <th>Bits</th> <th>Destination</th> </tr> </thead> <tbody> <tr> <td>10h</td> <td>GPP Phy interface block registers.</td> </tr> <tr> <td>20h</td> <td>GPP Phy registers.</td> </tr> <tr> <td>30h</td> <td>GPP Wrapper registers.</td> </tr> <tr> <td>40h</td> <td>GPP IO Link registers.</td> </tr> </tbody> </table>	Bits	Destination	10h	GPP Phy interface block registers.	20h	GPP Phy registers.	30h	GPP Wrapper registers.	40h	GPP IO Link registers.		
Bits	Destination												
10h	GPP Phy interface block registers.												
20h	GPP Phy registers.												
30h	GPP Wrapper registers.												
40h	GPP IO Link registers.												
15:0	<b>PcieIndxAddr: index address.</b> Read-write.												

**D0F0xE4 Link Index Data**

See D0F0xE0. Address: {D0F0xE0[BlockSelect],D0F0xE0[FrameType],D0F0xE0[PcieIndxAddr]}.

Bits	Description
31:0	<b>PcieIndxDatA: index data.</b>

**3.3.1 PIF Registers**

**D0F0xE4\_x0110\_0010 PIF Control**

Reset: 3190\_44D8h.

Bits	Description																				
31:20	Reserved.																				
19:17	<b>LS2ExitTime: LS2 exit time.</b> Read-write. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>14us</td> <td>100b</td> <td>30us</td> </tr> <tr> <td>001b</td> <td>10us</td> <td>101b</td> <td>100ns</td> </tr> <tr> <td>010b</td> <td>15us</td> <td>110b</td> <td>100us</td> </tr> <tr> <td>011b</td> <td>20us</td> <td>111b</td> <td>50us</td> </tr> </tbody> </table>	Bits	Definition	Bits	Definition	000b	14us	100b	30us	001b	10us	101b	100ns	010b	15us	110b	100us	011b	20us	111b	50us
Bits	Definition	Bits	Definition																		
000b	14us	100b	30us																		
001b	10us	101b	100ns																		
010b	15us	110b	100us																		
011b	20us	111b	50us																		
16:8	Reserved.																				
7	<b>RxDetectTxPwrMode: receiver detection transmitter power mode.</b> Read-write. 1=Transmitter is powered on.																				
6	<b>RxDetectFifoResetMode: receiver detect FIFO reset mode.</b> Read-write. BIOS: 1. 1=The transmit FIFO is reset after receiver detection. 0=The transmit FIFO is not reset after receiver detection.																				
5	Reserved.																				

4	<b>EiDetCycleMode: electrical idle detect mode.</b> 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\_x0110\_0011 PIF Pairing**

Reset: 0200\_0000h.

Bits	Description
31:26	Reserved.
25	<b>MultiPif: x16 link.</b> Read-write. 1=Lanes 7:0 are paired with a second PIF to create a x16 link.
24:21	Reserved.
20	<b>X16Lane150: x16 link lanes 15:0.</b> Read-write. 1=Lanes 15:0 are paired to create a x16 link.
19:18	Reserved.
17	<b>X8Lane158: x8 link lanes 15:8.</b> Read-write. 1=Lanes 15:8 are paired to create a x8 link.
16	<b>X8Lane70: x8 link lanes 7:0.</b> Read-write. 1=Lanes 7:0 are paired to create a x8 link.
15:12	Reserved.
11	<b>X4Lane1512: x4 link lanes 15:12.</b> Read-write. 1=Lanes 15:12 are paired to create a x4 link.
10	<b>X4Lane118: x4 link lanes 11:8.</b> Read-write. 1=Lanes 11:8 are paired to create a x4 link.
9	<b>X4Lane74: x4 link lanes 7:4.</b> Read-write. 1=Lanes 7:4 are paired to create a x4 link.
8	<b>X4Lane30: x4 link lanes 3:0.</b> Read-write. 1=Lanes 3:0 are paired to create a x4 link.
7	<b>X2Lane1514: x2 link lanes 15:14.</b> Read-write. 1=Lanes 15:14 are paired to create a x2 link
6	<b>X2Lane1312: x2 link lanes 13:12.</b> Read-write. 1=Lanes 13:12 are paired to create a x2 link
5	<b>X2Lane1110: x2 link lanes 11:10.</b> Read-write. 1=Lanes 11:10 are paired to create a x2 link
4	<b>X2Lane98: x2 link lanes 9:8.</b> Read-write. 1=Lanes 9:8 are paired to create a x2 link
3	<b>X2Lane76: x2 link lanes 7:6.</b> Read-write. 1=Lanes 7:6 are paired to create a x2 link.
2	<b>X2Lane54: x2 link lanes 5:4.</b> Read-write. 1=Lanes 5:4 are paired to create a x2 link.
1	<b>X2Lane32: x2 link lanes 3:2.</b> Read-write. 1=Lanes 3:2 are paired to create a x2 link.
0	<b>X2Lane10: x2 link lanes 1:0.</b> Read-write. 1=Lanes 1:0 are paired to create a x2 link.

**D0F0xE4\_x0110\_001[8:7,3:2] PIF Power Down Control [3:0]**

Reset: 0001\_1FA2h.

Table 89: Index addresses for **D0F0xE4\_x0110\_001[8:7,3:2]**

D0F0xE0[31:16]	D0F0xE0[15:0]			
	0018h	0017h	0013h	0012h
0110h	PIF Lanes 15-12	PIF Lanes 11-8	PIF Lanes 7-4	PIF Lanes 3-0

Bits	Description
31:29	<b>PIPwrOverrideVal: PLL power state override value.</b> Read-write. See TxPowerStateInTxs2.

28	<b>PlIPwrOverrideEn: PLL power state override enable.</b> Read-write. 1=PLL forced to the power state specified by PlIPwrOverrideVal.			
27	Reserved.			
26:24	<b>PlIRampUpTime: PLL ramp time.</b> Read-write.			
	<u>Bits</u>	<u>Definition</u>	<u>Bits</u>	<u>Definition</u>
	000b	10 us	100b	50 us
	001b	5 us	101b	300 us
	010b	15 us	110b	500 us
	011b	22 us	111b	800 us
23:17	Reserved.			
16	<b>Tx2p5clkClockGatingEn.</b> 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	<b>PlIPowerStateInOff: PLL off power state.</b> 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	<b>PlIPowerStateInTxs2: PLL L1 power state.</b> Read-write. See: TxPowerStateInTxs2. All links associated with the PLL must be in L1 to transition the PLL to this state.			
6:4	<b>RxPowerStateInRxs2: receiver L1 power state.</b> Read-write. See: TxPowerStateInTxs2.			
3	<b>ForceRxEnInL0s: force receiver enable in L0s.</b> Read-write. 1=The phy CDR is always enabled in L0s.			
2:0	<b>TxPowerStateInTxs2: transmitter L1 power state.</b> Read-write.			
	<u>Bits</u>	<u>Definition</u>	<u>Bits</u>	<u>Definition</u>
	000b	L0	100b	Reserved
	001b	LS1	101b	Reserved
	010b	LS2	110b	Reserved
	011b	Reserved	111b	Off

### 3.3.2 Phy Registers

#### 3.3.2.1 Global Phy Control Registers

##### D0F0xE4\_x0120\_0004 Phy Global Control 0

Bits	Description
31:18	Reserved.
17:16	<b>CfgIdleDetTh</b> . Read-write. Reset: 1. BIOS: 0. Idle detector threshold control
15:0	Reserved.

##### D0F0xE4\_x0120\_4440 Phy RO PLL Control

Bits	Description										
31:15	Reserved.										
14:13	<b>PllDbgRoIPFDResetCntrl</b> . Read-write. Reset: 0. BIOS: 2. PFD reset pulse width control. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>PFD reset pulse width = 603 ps (@85C_tt)</td> </tr> <tr> <td>01b</td> <td>PFD reset pulse width = 190 ps (@85C_tt)</td> </tr> <tr> <td>10b</td> <td>PFD reset pulse width = 397 ps (@85C_tt)</td> </tr> <tr> <td>11b</td> <td>PFD reset pulse width = 82 ps (@85C_tt)</td> </tr> </tbody> </table>	Bits	Definition	00b	PFD reset pulse width = 603 ps (@85C_tt)	01b	PFD reset pulse width = 190 ps (@85C_tt)	10b	PFD reset pulse width = 397 ps (@85C_tt)	11b	PFD reset pulse width = 82 ps (@85C_tt)
Bits	Definition										
00b	PFD reset pulse width = 603 ps (@85C_tt)										
01b	PFD reset pulse width = 190 ps (@85C_tt)										
10b	PFD reset pulse width = 397 ps (@85C_tt)										
11b	PFD reset pulse width = 82 ps (@85C_tt)										
12:0	Reserved.										

##### D0F0xE4\_x0120\_4450 PhyRO PLL Override Control 0

Bits	Description						
31	Reserved.						
30	<b>PllCfgROVTOIBiasCntrlOvrVal0</b> . Read-write. Reset: 1. BIOS: 0. Override value bit for CP_PLL_CFG_RO_VTOI_BIAS_CNTRL.						
29:8	Reserved.						
7:0	<b>PllCfgROBWCntrlOvrVal0</b> . Read-write. Reset: 8Dh. BIOS: 90h. Override value bit for CP_PLL_CFG_RO_BW_CNTRL. <table border="1"> <thead> <tr> <th>Bit</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>[7:3]</td> <td>Proportional CP current setting</td> </tr> <tr> <td>[2:0]</td> <td>Integral CP current setting</td> </tr> </tbody> </table>	Bit	Definition	[7:3]	Proportional CP current setting	[2:0]	Integral CP current setting
Bit	Definition						
[7:3]	Proportional CP current setting						
[2:0]	Integral CP current setting						

### 3.3.3 Wrapper Registers

#### D0F0xE4\_x0130\_0046 Subsystem and Vendor ID

Bits	Description
31:16	<b>SubsystemID: subsystem id.</b> Read-write. Reset: 1234h. Specifies the value returned by <a href="#">D2F[5:1]xB4[SubsystemID]</a> .
15:0	<b>SubsystemVendorID: subsystem vendor id .</b> Read-write. Reset: 1022h. Specifies the value returned by <a href="#">D2F[5:1]xB4[SubsystemVendorID]</a> .

#### D0F0xE4\_x0130\_0080 Link Configuration

Reset: 0000\_0000h.

Bits	Description																				
31:4	Reserved.																				
3:0	<p><b>StrapBifLinkConfig.</b> Read-write; strap. Reset: Product-specific.                      BIOS: See <a href="#">Table 62</a>.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>0000b</td> <td>Reserved</td> <td>0100b</td> <td>4 x1 IO Links (GPPFCH Only)</td> </tr> <tr> <td>0001b</td> <td>x4 IO Link (GPPFCH Only)</td> <td>0101b</td> <td>Reserved</td> </tr> <tr> <td>0010b</td> <td>2 x2 IO Links (GPPFCH Only)</td> <td>011xb</td> <td>Reserved</td> </tr> <tr> <td>0011b</td> <td>1 x2 IO Link, 2 x1 IO Links (GPPFCH Only)</td> <td>1xxx b</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Definition	Bits	Definition	0000b	Reserved	0100b	4 x1 IO Links (GPPFCH Only)	0001b	x4 IO Link (GPPFCH Only)	0101b	Reserved	0010b	2 x2 IO Links (GPPFCH Only)	011xb	Reserved	0011b	1 x2 IO Link, 2 x1 IO Links (GPPFCH Only)	1xxx b	Reserved
Bits	Definition	Bits	Definition																		
0000b	Reserved	0100b	4 x1 IO Links (GPPFCH Only)																		
0001b	x4 IO Link (GPPFCH Only)	0101b	Reserved																		
0010b	2 x2 IO Links (GPPFCH Only)	011xb	Reserved																		
0011b	1 x2 IO Link, 2 x1 IO Links (GPPFCH Only)	1xxx b	Reserved																		

#### D0F0xE4\_x0130\_0[C:8]00 Link Hold Training Control

Table 90: Index address mapping for [D0F0xE4\\_x0130\\_0\[C:8\]00](#)

Index	Function
0130_0800h	GPPSB PortA
0130_0900h	GPPSB PortB
0130_0A00h	GPPSB PortC
0130_0B00h	GPPSB PortD
0130_0C00h	GPPSB PortE

Bits	Description
31:1	Reserved.
0	<b>HoldTraining: hold link training.</b> Read-write. Reset: 1. 1=Hold training on link.

**D0F0xE4\_x0130\_0[C:8]03 Link Deemphasis Control**

Table 91: Index address mapping for D0F0xE4\_x0130\_0[C:8]03

Index	Function
0130_0803h	GPPSB PortA
0130_0903h	GPPSB PortB
0130_0A03h	GPPSB PortC
0130_0B03h	GPPSB PortD
0130_0C03h	GPPSB PortE

Bits	Description
31:6	Reserved.
5	<b>StrapBifDeemphasisSel.</b> Read-write; strap. Reset: 1. Controls the default value of <a href="#">D2F[5:1]x88[SelectableDeemphasis]</a> . 1=RC advertises -3.5dB. 0=RC advertises -6dB.
4:0	Reserved.

**D0F0xE4\_x0130\_0[C:8]04 Link AER Control**

Bits	Description
31:4	Reserved.
3:1	<b>StrapBifMaxPayloadSupport.</b> Read-write; strap. Reset: 2h. BIOS: See <a href="#">2.11.3.2 [Link Configurations]</a> .
0	Reserved.

**D0F0xE4\_x0130\_0[C:8]05 Link Training Control**

Reset: 1800\_0000h.

Table 92: Index address mapping for D0F0xE4\_x0130\_0[C:8]05

Index	Function
0130_0805h	GPPSB PortA
0130_0905h	GPPSB PortB
0130_0A05h	GPPSB PortC
0130_0B05h	GPPSB PortD
0130_0C05h	GPPSB PortE

Bits	Description
31:24	<b>StrapBifInitialNFts.</b> Read-write; strap. BIOS: 40h. Specifies the number of Fast Training Sets(FTS) transmitted. FTS are special packets transmitted to exit out of the PCIe® L0s sleep state.
23:0	Reserved.



**D0F0xE4\_x0130\_8002 IO Link Wrapper Scratch**

Cold reset: 0000\_0000h.

Bits	Description
31:0	<b>PcieWrapScratch: Scratch.</b> Read-write.

**D0F0xE4\_x0130\_8011 Link Transmit Clock Gating Control**

Bits	Description
31:26	Reserved.
25	Reserved.
24	<b>TxclkLcntGateEnable.</b> Read-write. Reset: 0. BIOS: 1. 1=Enable clock gating the lane counter.
23	<b>DebugBusClkEnable.</b> Read-write. Reset: 1. BIOS: 0. 1=Enable the debug bus clock.
22:17	<b>TxclkPermGateLatency.</b> Read-write. Reset: 3Fh. Specifies the number of clocks to wait after detecting an entry into L1 before gating off the permanent clock branches.
16	<b>RcvrDetClkEnable.</b> Read-write. Reset: 0. 1=Enable the receiver detect clock.
15:10	<b>TxclkRegsGateLatency.</b> Read-write. Reset: 3Fh. Specifies the number of clocks to wait after idle is signalled before gating off the register clock branch.
9	<b>TxclkRegsGateEnable.</b> Read-write. Reset: 0. BIOS: 1. 1=Enable clock gating the register clock.
8	<b>TxclkPermStop.</b> 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.
7	<b>TxclkDynGateEnable.</b> Read-write. Reset: 0. BIOS: 1. 1=Dynamic clock gating enabled. 0=Dynamic clock gating disabled.
6	<b>TxclkPermGateEven.</b> Read-write. Reset: 1. 1=Gate the permanent clock branches for an even number of clocks.
5:0	<b>TxclkDynGateLatency.</b> Read-write. Reset: 3Fh. Specifies the number of clocks to wait after idle is signalled before gating off the dynamic clock branch.

**D0F0xE4\_x0130\_8012 Link Idle-Resume Clock Gating Control**

Bits	Description
31:14	Reserved.
13:8	<b>Pif1xIdleResumeLatency.</b> Read-write. Reset: 00_0111b. Specifies the number of clocks to wait after enabling TXCLK1X_PIF before sending the acknowledge.
7	<b>Pif1xIdleGateEnable.</b> Read-write. Reset: 0. BIOS: 1. 1=Enable idle resume gating of TXCLK1X_PIF.
6	Reserved.
5:0	<b>Pif1xIdleGateLatency.</b> Read-write. Reset: 0_0001b. Specifies the number of clocks to wait before turning off TXCLK1X_PIF.

**D0F0xE4\_x0130\_8013 Transmit Clock Pll Control**

Reset: 0000\_0001h.

Bits	Description
31:15	Reserved.
14:13	<b>PhyRxIsoDis.</b> Read-write. 1=Isolate PHY signals to PIF.
12	Reserved.
11	Reserved.
10	<b>TxclkSelPifBOverride.</b> Read-write. 1=Override TxclkPifB selection.
9	<b>TxclkSelPifAOverride.</b> Read-write. 1=Override TxclkPifA selection.
8	<b>TxclkSelCoreOverride.</b> Read-write. 1=Override TxclkCore selection.
7	Reserved.
6	Reserved.
5	<b>ClkDividerResetOverrideB.</b> Read-write. 1=Force clock divider B enabled.
4	<b>ClkDividerResetOverrideA.</b> Read-write. 1=Force clock divider A enabled.
3	Reserved.
2	Reserved.
1	<b>MasterPciePIIB.</b> Read-write. 1=Pll B is the master source for all PCIe® transmitter clock branches.
0	<b>MasterPciePIIA.</b> Read-write. 1=Pll A is the master source for all PCIe® transmitter clock branches.

**D0F0xE4\_x0130\_8014 Link Transmit Clock Gating Control 2**

Reset: 0000\_0000h.

Bits	Description
31:28	<b>SpareRegRw.</b> Read-write. Spare register.
27	Reserved.
26	Reserved.
25	Reserved.
24	Reserved.

23:21	Reserved.
20	<b>TxclkPermGateOnlyWhenPllPwrDn.</b> Read-write. BIOS: 1. 1=Gating of the permanent clock branch only occurs when the PLL is powered down.
19:16	Reserved.
15	Reserved.
14	Reserved.
13	<b>PcieGatePifB1xEnable.</b> Read-write. BIOS: 1. 1=Enable gating of the PIF B 1x clock branches in PCIe <sup>®</sup> mode.
12	<b>PcieGatePifA1xEnable.</b> Read-write. BIOS: 1. 1=Enable gating of the PIF A 1x clock branches in PCIe <sup>®</sup> mode.
11:6	Reserved.
5	Reserved.
4	Reserved.
3	Reserved.
2	Reserved.
1	<b>TxclkPrbsGateEnable.</b> Read-write. BIOS: 1. 1=Enable gating of the PRBS clock branch.
0	<b>TxclkPermGateEnable.</b> Read-write. BIOS: 1. 1=Enable gating of the permanent clock branch.

#### **D0F0xE4\_x0130\_8015 IO Link IOC Control**

Bits	Description
31:24	Reserved.
23	<b>RefclkRegsGateEnable.</b> Read-write. Reset:0. BIOS: 1. 1=Enable gating of REFCLK_REGS.
22	Reserved.
21:16	<b>RefclkRegsGateLatency.</b> Read-write. Reset:3Fh. Specifies the number of clocks to wait before turning of f REFCLK_REGS.
15:0	Reserved.

#### **D0F0xE4\_x0130\_8016 Link Clock Switching Control**

Reset: 003F\_001Fh.

Bits	Description
31:24	Reserved.
23	<b>LclkDynGateEnable.</b> Read-write. 1=Enable LCLK_DYN clock gating.

22	<b>LclkGateFree</b> . Read-write. IF (REG== D0F0xE4_x013[1:0]_8016) THEN BIOS: 1. ENDIF. 1=LCLK gating is controlled independent of TXCLK gating.
21:16	<b>LclkDynGateLatency</b> . Read-write. Specifies the number of clocks to wait before turning off LCLK_DYN.
15:6	Reserved.
5:0	<b>CalibAckLatency</b> . Read-write. Specifies the number of clocks after calibration is complete before the acknowledge signal is asserted.

**D0F0xE4\_x0130\_802[4:1] Transmitter Lane Mux**

Table 93: Lane index addresses for [D0F0xE4\\_x0130\\_802\[4:1\]](#)

D0F0xE0[31:4]	D0F0xE0[3:0]			
	4h	3h	2h	1h
<a href="#">0130_802h</a>	Lanes[15:12]	Lanes[11:8]	Lanes[7:4]	Lanes[3:0]

**Table 94:** Reset Mapping for [D0F0xE4\\_x0130\\_802\[4:1\]](#)

Register	Reset
D0F0xE4_x0130_8024	0F0E_0D0Ch
D0F0xE4_x0130_8023	0B0A_0908h
D0F0xE4_x0130_8022	0706_0504h
D0F0xE4_x0130_8021	0302_0100h

Table 95: Field mapping for [D0F0xE4\\_x0130\\_802\[4:1\]](#)

Register	Bits			
	31:24	23:16	15:8	7:0
D0F0xE4_x0130_8024	TXLane15	TXLane14	TXLane13	TXLane12
D0F0xE4_x0130_8023	TXLane11	TXLane10	TXLane9	TXLane8
D0F0xE4_x0130_8022	TXLane7	TXLane6	TXLane5	TXLane4
D0F0xE4_x0130_8021	TXLane3	TXLane2	TXLane1	TXLane0

Bits	Description
31:24	<b>TXLane</b> . Read-write. Specifies the controller lanes that are mapped to TX lane n of the PIF. See: <a href="#">D0F0xE4_x0130_802[4:1][7:0]</a> .
23:16	<b>TXLane</b> . Read-write. Specifies the controller lanes that are mapped to TX lane n of the PIF. See: <a href="#">D0F0xE4_x0130_802[4:1][7:0]</a> .

15:8	<b>TXLane.</b> Read-write. Specifies the controller lanes that are mapped to TX lane n of the PIF. See: <a href="#">D0F0xE4_x0130_802[4:1][7:0]</a> .																																				
7:0	<b>TXLane.</b> Read-write. Specifies the controller lanes that are mapped to TX lane n of the PIF. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Controller lane 0.</td> <td>8h</td> <td>Controller lane 8.</td> </tr> <tr> <td>1h</td> <td>Controller lane 1.</td> <td>9h</td> <td>Controller lane 9.</td> </tr> <tr> <td>2h</td> <td>Controller lane 2.</td> <td>10h</td> <td>Controller lane 10.</td> </tr> <tr> <td>3h</td> <td>Controller lane 3.</td> <td>11h</td> <td>Controller lane 11.</td> </tr> <tr> <td>4h</td> <td>Controller lane 4.</td> <td>12h</td> <td>Controller lane 12.</td> </tr> <tr> <td>5h</td> <td>Controller lane 5.</td> <td>13h</td> <td>Controller lane 13.</td> </tr> <tr> <td>6h</td> <td>Controller lane 6.</td> <td>14h</td> <td>Controller lane 14.</td> </tr> <tr> <td>7h</td> <td>Controller lane 7.</td> <td>15h</td> <td>Controller lane 15.</td> </tr> </tbody> </table>	Bits	Definition	Bits	Definition	0h	Controller lane 0.	8h	Controller lane 8.	1h	Controller lane 1.	9h	Controller lane 9.	2h	Controller lane 2.	10h	Controller lane 10.	3h	Controller lane 3.	11h	Controller lane 11.	4h	Controller lane 4.	12h	Controller lane 12.	5h	Controller lane 5.	13h	Controller lane 13.	6h	Controller lane 6.	14h	Controller lane 14.	7h	Controller lane 7.	15h	Controller lane 15.
Bits	Definition	Bits	Definition																																		
0h	Controller lane 0.	8h	Controller lane 8.																																		
1h	Controller lane 1.	9h	Controller lane 9.																																		
2h	Controller lane 2.	10h	Controller lane 10.																																		
3h	Controller lane 3.	11h	Controller lane 11.																																		
4h	Controller lane 4.	12h	Controller lane 12.																																		
5h	Controller lane 5.	13h	Controller lane 13.																																		
6h	Controller lane 6.	14h	Controller lane 14.																																		
7h	Controller lane 7.	15h	Controller lane 15.																																		

**D0F0xE4\_x0130\_802[8:5] Receiver Lane Mux**

Reset: 0302\_0100h.

Table 96: Lane index addresses for [D0F0xE4\\_x0130\\_802\[8:5\]](#)

D0F0xE0[31:4]	D0F0xE0[3:0]			
	8h	7h	6h	5h
<a href="#">0130_802h</a>	Lanes[15:12]	Lanes[11:8]	Lanes[7:4]	Lanes[3:0]

Table 97: [Reset Mapping](#) for [D0F0xE4\\_x0130\\_802\[8:5\]](#)

Register	Reset
D0F0xE4_x0130_8028	0F0E_0D0Ch
D0F0xE4_x0130_8027	0B0A_0908h
<a href="#">D0F0xE4_x0130_8026</a>	0706_0504h
D0F0xE4_x0130_8025	0302_0100h

Table 98: Field mapping for [D0F0xE4\\_x0130\\_802\[8:5\]](#)

Register	Bits			
	31:24	23:16	15:8	7:0
D0F0xE4_x0130_8028	RXLane15	RXLane14	RXLane13	RXLane12
D0F0xE4_x0130_8027	RXLane11	RXLane10	RXLane9	RXLane8
D0F0xE4_x0130_8026	RXLane7	RXLane6	RXLane5	RXLane4
D0F0xE4_x0130_8025	RXLane3	RXLane2	RXLane1	RXLane0

Bits	Description
31:24	<b>RXLane.</b> Read-write. Specifies the PIF RX lanes that are mapped to controller lane n. See: <a href="#">D0F0xE4_x0130_802[8:5][7:0]</a> .
23:16	<b>RXLane.</b> Read-write. Specifies the PIF RX lanes that are mapped to controller lane n. See: <a href="#">D0F0xE4_x0130_802[8:5][7:0]</a> .

15:8	<b>RXLane</b> . Read-write. Specifies the PIF RX lanes that are mapped to controller lane n. See: <a href="#">D0F0xE4_x0130_802[8:5][7:0]</a> .			
7:0	<b>RXLane</b> . Read-write. Specifies the PIF RX lanes that are mapped to controller lane n.			
	<u>Bits</u>	<u>Definition</u>	<u>Bits</u>	<u>Definition</u>
	0h	PIF RX lane 0.	8h	PIF RX lane 8.
	1h	PIF RX lane 1.	9h	PIF RX lane 9.
	2h	PIF RX lane 2.	10h	PIF RX lane 10.
	3h	PIF RX lane 3.	11h	PIF RX lane 11.
	4h	PIF RX lane 4.	12h	PIF RX lane 12.
	5h	PIF RX lane 5.	13h	PIF RX lane 13.
	6h	PIF RX lane 6.	14h	PIF RX lane 14.
	7h	PIF RX lane 7.	15h	PIF RX lane 15.

**D0F0xE4\_x0130\_8029 Lane Enable**

Reset: 0000\_FFFFh.

Bits	Description
31:16	Reserved.
15:0	<b>LaneEnable</b> . Read-write. 1=Lane enabled for transmit.
	<u>Bit</u> <u>Definition</u>
	[15:0]    Lane <BIT> enable

**D0F0xE4\_x0130\_8060 Soft Reset Command 0**

Cold reset: 0000\_0000h.

Bits	Description
31:18	Reserved.
17	<b>Bif0CalibrationReset</b> . Read-write. 1=The BIF 0 calibration block reset is asserted.
16	<b>Bif0GlobalReset</b> . Read-write. 1=The BIF 0 global reset is asserted.
15:4	Reserved.
3	<b>WaitState</b> . Read-only. 1=Reset cycle is in the wait state.
2	<b>ResetComplete</b> . Read-only. 1=Reset cycle is complete.
1	Reserved.
0	<b>Reconfigure</b> . Read-write; Cleared-when-done. 1=Trigger atomic reconfiguration if <a href="#">D0F0xE4_x0130_8062[ReconfigureEn]=1</a> .

**D0F0xE4\_x0130\_8062 Soft Reset Control 0**

Cold reset: 0001\_0880h.

Bits	Description
31:12	Reserved.
11	<b>ConfigXferMode</b> . Read-write. 1=PCIe® core strap settings take effect immediately. 0=PCIe® core strap settings take effect when the PCIe® core is reset.

10	<b>BlockOnIdle.</b> Read-write. 1=The PCIe <sup>®</sup> core must be idle before hardware initiates a reconfiguration. 0=The PCIe <sup>®</sup> core does not have to be idle before hardware initiates a reconfiguration.
9:5	Reserved.
4:2	<b>ResetPeriod.</b> Read-write. BIOS:2h. Specifies the amount of time that resets are asserted during a reconfiguration. 5h-7h: Reserved.
1	Reserved.
0	<b>ReconfigureEn.</b> Read-write. 1=Atomic reconfiguration enabled.

**D0F0xE4\_x0130\_80F0 BIOS Timer**

Reset: 0000\_0000h.

Bits	Description
31:0	<b>MicroSeconds.</b> Read-write; Updated-by-hardware. This field increments once every microsecond when the timer is enabled. The counter rolls over and continues 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	<b>ClockRate.</b> Read-write. Specifies the frequency of the reference clock in 1 MHz increments. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00h</td> <td>Timer disabled</td> </tr> <tr> <td>FFh-01h</td> <td>&lt;ClockRate&gt; MHz</td> </tr> </tbody> </table>	Bits	Definition	00h	Timer disabled	FFh-01h	<ClockRate> MHz
Bits	Definition						
00h	Timer disabled						
FFh-01h	<ClockRate> MHz						

**3.3.4 IO Link Registers****D0F0xE4\_x0140\_0002 IO Link Hardware Debug**

Reset: 0000\_0000h.

Bits	Description
31:1	Reserved.
0	<b>HwDebug[0]: ignore DLLPs in L1.</b> Read-write. BIOS: 1. 1=DLLPs are ignored in L1 so the TXCLK can be turned off.

**D0F0xE4\_x0140\_0010 IO Link Control 1**

Reset: 80E3\_110Bh.

Bits	Description
31:13	Reserved.

12:10	<b>RxUmiAdjPayloadSize.</b> Read-write. BIOS: 100b. Payload size for the UMI link. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00xb</td> <td>Reserved.</td> <td>100b</td> <td>64 bytes</td> </tr> <tr> <td>010b</td> <td>16 bytes</td> <td>101b</td> <td>Reserved.</td> </tr> <tr> <td>011b</td> <td>32 bytes.</td> <td>11xb</td> <td>Reserved.</td> </tr> </tbody> </table>	Bits	Definition	Bits	Definition	00xb	Reserved.	100b	64 bytes	010b	16 bytes	101b	Reserved.	011b	32 bytes.	11xb	Reserved.				
Bits	Definition	Bits	Definition																		
00xb	Reserved.	100b	64 bytes																		
010b	16 bytes	101b	Reserved.																		
011b	32 bytes.	11xb	Reserved.																		
9	<b>UmiNpMemWrite: memory write mapping enable.</b> Read-write. 1=Internal non-posted memory writes are transferred to UMI.																				
8:4	Reserved.																				
3:1	<b>LcHotPlugDelSel: enhanced hot plug counter select.</b> Read-write. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>15 ms</td> <td>4h</td> <td>150 ms</td> </tr> <tr> <td>1h</td> <td>20 ms</td> <td>5h</td> <td>200 ms</td> </tr> <tr> <td>2h</td> <td>50 ms</td> <td>6h</td> <td>275 ms</td> </tr> <tr> <td>3h</td> <td>100 ms</td> <td>7h</td> <td>335 ms</td> </tr> </tbody> </table>	Bits	Definition	Bits	Definition	0h	15 ms	4h	150 ms	1h	20 ms	5h	200 ms	2h	50 ms	6h	275 ms	3h	100 ms	7h	335 ms
Bits	Definition	Bits	Definition																		
0h	15 ms	4h	150 ms																		
1h	20 ms	5h	200 ms																		
2h	50 ms	6h	275 ms																		
3h	100 ms	7h	335 ms																		
0	<b>HwInitWrLock: hardware init write lock.</b> Read-write. 1=Lock HWInit registers. 0=Unlock HWInit registers.																				

**D0F0xE4\_x0140\_0011 IO Link Config Control**

Reset: 0000\_000Fh.

Bits	Description
31:4	Reserved.
3:0	<b>DynClkLatency: dynamic clock latency.</b> Read-write. BIOS: See <a href="#">2.11.4.3.1 [Link Configuration and Core Initialization]</a> . Specifies the number of clock cycles after logic goes idle before clocks are gated off.

**D0F0xE4\_x0140\_001C IO Link Control 2**

Reset: 0E00\_0109h.

Bits	Description
31:11	Reserved.
10:6	<b>TxArbMstLimit: transmitter arbitration master limit.</b> 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	<b>TxArbSlvLimit: transmitter arbitration slave limit.</b> Read-write. BIOS: 4h. See TxArbMstLimit for details
0	<b>TxArbRoundRobinEn: transmitter round robin arbitration enabled.</b> Read-write. BIOS: 1. 1=Enable transmitter round robin arbitration. 0=Disable transmitter round robin arbitration.



**D0F0xE4\_x0140\_0020 IO Link Chip Interface Control**

Reset: 0000\_0050h.

Bits	Description
31:10	Reserved.
9	<b>CiRcOrderingDis: chip interface RC ordering disable.</b> Read-write. 0=RC ordering logic is enabled. 1=RC ordering logic is disabled.
8:0	Reserved.

**D0F0xE4\_x0140\_0040 IO Link Phy Control**

Reset: 0001\_0000h.

Bits	Description										
31:16	Reserved.										
15:14	<b>PElecIdleMode: electrical idle mode for physical layer.</b> Read-write. BIOS: 01b. Defines which electrical idle signal is used, either inferred by link controller or from phy. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Gen1 - entry:PHY, exit:PHY; Gen2 - entry:INF, exit:PHY.</td> </tr> <tr> <td>01b</td> <td>Gen1 - entry:INF, exit:PHY; Gen2 - entry:INF, exit:PHY.</td> </tr> <tr> <td>10b</td> <td>Gen1 - entry:PHY, exit:PHY; Gen2 - entry:PHY, exit:PHY.</td> </tr> <tr> <td>11b</td> <td>Gen1 - entry: PHY, exit: PHY; Gen2 - entry: PHY, exit: PHY.</td> </tr> </tbody> </table>	Bits	Definition	00b	Gen1 - entry:PHY, exit:PHY; Gen2 - entry:INF, exit:PHY.	01b	Gen1 - entry:INF, exit:PHY; Gen2 - entry:INF, exit:PHY.	10b	Gen1 - entry:PHY, exit:PHY; Gen2 - entry:PHY, exit:PHY.	11b	Gen1 - entry: PHY, exit: PHY; Gen2 - entry: PHY, exit: PHY.
Bits	Definition										
00b	Gen1 - entry:PHY, exit:PHY; Gen2 - entry:INF, exit:PHY.										
01b	Gen1 - entry:INF, exit:PHY; Gen2 - entry:INF, exit:PHY.										
10b	Gen1 - entry:PHY, exit:PHY; Gen2 - entry:PHY, exit:PHY.										
11b	Gen1 - entry: PHY, exit: PHY; Gen2 - entry: PHY, exit: PHY.										
13:0	Reserved.										

**D0F0xE4\_x0140\_00B0 IO Link Strap Control**

Reset: 0000\_8001h.

Bits	Description
31:6	Reserved.
5	<b>StrapF0AerEn.</b> Read-write. 1=AER support enabled. 0=AER support disabled.
4:3	Reserved.
2	<b>StrapF0MsiEn.</b> Read-write. BIOS: 1. Overrides MSI enable.
1:0	Reserved.

**D0F0xE4\_x0140\_00C0 IO Link Strap Miscellaneous**

Bits	Description
31	Reserved.
30	<b>StrapFlrEn.</b> Read-write.
29	<b>StrapMstAdr64En.</b> Read-write. Reset: 0.
28	<b>StrapReverseAll.</b> Read-write. Reset: 0.
27:0	Reserved.

**D0F0xE4\_x0140\_00C1 IO Link Strap Miscellaneous2**

Bits	Description
31:2	Reserved.
1	<b>StrapGen2Compliance</b> . Read-write. Reset: 1.
0	<b>StrapLinkBwNotificationCapEn</b> . Read-write. Reset: 0.

**D0F0xF8 Northbridge IOAPIC Index**

Reset: 0000\_0000h. The index/data pair registers, [D0F0xF8](#) and [D0F0xFC](#), are used to access the registers at [D0F0xFC\\_x\[FF:00\]](#). To access any of these registers, the address is first written into the index register, [D0F0xF8](#), and then the data is read from or written to the data register, [D0F0xFC](#).

Bits	Description
31:8	Reserved.
7:0	<b>IOAPICIndAddr: IOAPIC index register address</b> . Read-write.

**D0F0xFC Northbridge IOAPIC Data**

Reset: 0000\_0000h. See [D0F0xF8](#). Address: [D0F0xF8](#)[[IOAPICIndAddr](#)].

Bits	Description
31:0	<b>IOAPICIndData: IOAPIC index data register</b> . Read-write.

**D0F0xFC\_x00 IOAPIC Feature Control Register**

Reset: 0000\_0004h.

Bits	Description
31:5	Reserved.
4	<b>IoapicSbFeatureEn</b> . Read-write. 1=Enable masked interrupts to be routed back to the FCH PIC/IOAPIC.
3	Reserved.
2	<b>IoapicIdExtEn</b> . Read-write. Extend the IOAPIC ID from 4-bit to 8-bit. 0=4-bit ID. 1=8-bit ID.
1	Reserved.
0	<b>IoapicEnable</b> . Read-write. 1=Enables the INTGEN block to decode IOAPIC addresses. BIOS: 1. BIOS should always set this bit after programming the IOAPIC BAR in the init sequence.

**D0F0xFC\_x01 IOAPIC Base Address Lower**

Reset: FEC0\_0000h. See [3.15 \[Northbridge IOAPIC Registers\]](#).

Bits	Description
31:8	<b>IoapicAddr</b> . Read-write. IOAPIC Base Address bits [31:8].
7:0	Reserved.

**D0F0xFC\_x02 IOAPIC Base Address Upper**

Reset: 0000\_0000h. See 3.15 [Northbridge IOAPIC Registers].

Bits	Description
31:0	<b>IoapicAddrUpper.</b> Read-write. IOAPIC Base Address bits [63:32].

**D0F0xFC\_x0F IOAPIC GBIF Interrupt Routing Register**

Reset: 0000\_0000h.

Bits	Description										
31:6	Reserved.										
5:4	<b>GBIFExtIntrSwz.</b> Read-write. Swizzle GBIF INTA/B/C/D based on the value in this field before mapping them onto the IOAPIC pins. <table border="1"> <thead> <tr> <th>Bits</th> <th>Interrupt Swizzling</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>ABCD</td> </tr> <tr> <td>01b</td> <td>BCDA</td> </tr> <tr> <td>10b</td> <td>CDAB</td> </tr> <tr> <td>11b</td> <td>DABC</td> </tr> </tbody> </table>	Bits	Interrupt Swizzling	00b	ABCD	01b	BCDA	10b	CDAB	11b	DABC
Bits	Interrupt Swizzling										
00b	ABCD										
01b	BCDA										
10b	CDAB										
11b	DABC										
3	Reserved.										
2:0	<b>GBIFExtIntrGrp.</b> Read-write. Map GBIF INTA/B/C/D to IOAPIC pins $[(grp+1)*4-1:(grp*4)]$ . For GBIF, only INTA/B are used. INTC/D should be tied off.										

**D0F0xFC\_x1[4:0] IOAPIC BR Interrupt Routing Register**

Reset: 0000\_0000h.

Bits	Description										
31:21	Reserved.										
20:16	<b>BrIntIntrMap.</b> Read-write. Map bridge n interrupts to IOAPIC redirection table entry.										
15:6	Reserved.										
5:4	<b>BrExtIntrSwz.</b> Read-write. Swizzle bridge n external INTA/B/C/D based on the value in this field before mapping them onto the IOAPIC pins. <table border="1"> <thead> <tr> <th>Bits</th> <th>Interrupt Swizzling</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>ABCD</td> </tr> <tr> <td>01b</td> <td>BCDA</td> </tr> <tr> <td>10b</td> <td>CDAB</td> </tr> <tr> <td>11b</td> <td>DABC</td> </tr> </tbody> </table>	Bits	Interrupt Swizzling	00b	ABCD	01b	BCDA	10b	CDAB	11b	DABC
Bits	Interrupt Swizzling										
00b	ABCD										
01b	BCDA										
10b	CDAB										
11b	DABC										
3	Reserved.										
2:0	<b>BrExtIntrGrp.</b> Read-write. Map bridge n external INTA/B/C/D to IOAPIC pins $[(grp+1)*4-1:(grp*4)]$ .										

**D0F0xFC\_x30 IOAPIC Serial IRQ Status**

---

Reset: 0000\_0000h.

Bits	Description
31:0	<b>InternalIrqSts.</b> Read-only. Shows the status of the 32 IOAPIC interrupt pins.

**D0F0xFC\_x3[F:E] IOAPIC Scratch [1:0] Register**

---

Reset: 0000\_0000h.

Bits	Description
31:0	<b>Scratch.</b> Read-write.

### 3.4 Device 0 Function 2 Configuration Registers

See 3.1 [Register Descriptions and Mnemonics]. See 2.7 [Configuration Space]. See 2.11.6 [PCIe® Client Interface Control].

#### **D0F2xF8 PCIe® Client Interface Index**

The index/data pair registers, **D0F2xF8** and **D0F2xFC** are used to access the registers at **D0F2xFC\_x[FFFF:0000]\_L1i[3:0]**. To access any of these registers, the address is first written into the index register, **D0F2xF8**, and then the data is read from or written to the data register, **D0F2xFC**.

Registers in the L1 indexed space have one instance per L1 denoted by **\_L1i[x]** where  $x = \text{D0F2xF8}[\text{L1cfgSel}]$ . The syntax for this register type is described by example as follows:

- **D0F2xFC\_x32** refers to all instances of the **D0F2xFC\_x32\_L1i** registers.
- **D0F2xFC\_x32\_L1i\_L1i[1]** refers to the **D0F2xFC\_x32** register instance for the BIF L1.

Bits	Description										
31	<b>L1cfgEn</b> . Read-write. Reset: 0. 1=Enable writes to <b>D0F2xFC</b> .										
30:20	Reserved.										
19:16	<b>L1cfgSel</b> . Read-write. Reset: 0. This field selects one of the four L1s to access. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>PPD L1</td> </tr> <tr> <td>1h</td> <td>BIF L1</td> </tr> <tr> <td>2h</td> <td>INTGEN L1</td> </tr> <tr> <td>Fh-3h</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Description	0h	PPD L1	1h	BIF L1	2h	INTGEN L1	Fh-3h	Reserved
Bits	Description										
0h	PPD L1										
1h	BIF L1										
2h	INTGEN L1										
Fh-3h	Reserved										
15:0	<b>L1cfgIndex</b> . Read-write. Reset: 0.										

#### **D0F2xFC PCIe® Client Interface Data**

IF (**D0F2xF8**[L1cfgEn]) THEN Read-write. ELSE Read-only. ENDIF. Reset: 0000\_0000h. See **D0F2xF8**. Address: **D0F2xF8**[L1cfgIndex].

Bits	Description
31:0	<b>L1cfgData</b> .

#### **D0F2xFC\_x32\_L1i L1\_CNTRL\_4**

Bits	Description
31:20	Reserved.
19:17	Reserved.
16	<b>DmaNpHaltDis</b> . Read-write. Reset: 0.
15:10	<b>DmaBufMaxNpCred</b> . Read-write. Reset: Fh.
9:4	<b>DmaBufCredits</b> . Read-write. Reset: 10h.
3	Reserved.

2	Reserved.
1	Reserved.
0	Reserved.

**D0F2xFC\_x33\_L1i L1\_CLKCNTRL\_0**

Bits	Description
31	<b>L1L2ClkgateEn.</b> Read-write. Reset: 0. BIOS: 1.
30:12	Reserved.
11	<b>L1HostreqClkgateEn.</b> Read-write. Reset: 0. BIOS: 1.
10	<b>L1RegClkgateEn.</b> Read-write. Reset: 0. BIOS: 1.
9	<b>L1MemoryClkgateEn.</b> Read-write. Reset: 0. BIOS: 1.
8	<b>L1PerfClkgateEn.</b> Read-write. Reset: 0. BIOS: 1.
7	<b>L1DmaInputClkgateEn.</b> Read-write. Reset: 0. BIOS: 1.
6	<b>L1CpslvClkgateEn.</b> Read-write. Reset: 0. BIOS: 1.
5	<b>L1CacheClkgateEn.</b> Read-write. Reset: 0. BIOS: 1.
4	<b>L1DmaClkgateEn.</b> Read-write. Reset: 0. BIOS: 1.
3:2	Reserved.
1:0	<b>L1ClkgateLen.</b> Read-write. Reset: 0.

### 3.5 Device 1 Function 0 (Internal Graphics) Configuration Registers

See 3.1 [Register Descriptions and Mnemonics]. See 2.7 [Configuration Space].

#### D1F0x00 Device/Vendor ID

Bits	Description
31:16	<b>DeviceID: device ID.</b> Read-only. Value: Product-specific. .
15:0	<b>VendorID: vendor ID.</b> Read-only. Value: 1002h.

#### D1F0x04 Status/Command Register

Reset: 0010\_0000h.

Bits	Description
31	<b>ParityErrorDetected: detected parity error.</b> Read; Write-1-to-clear. 1=Poisoned TLP received.
30	<b>SignaledSystemError: signaled system error.</b> Read; Write-1-to-clear. 1=A non-fatal or fatal error message was sent and SerrEn=1.
29	<b>ReceivedMasterAbort: received master abort.</b> Read; Write-1-to-clear. 1=A completion with an unsupported request completion status was received.
28	<b>ReceivedTargetAbort: received target abort.</b> Read; Write-1-to-clear. 1=A completion with completer abort completion status was received.
27	<b>SignalTargetAbort: Signaled target abort.</b> Read-only.
26:25	<b>DevselTiming: DEVSEL# Timing.</b> Read-only.
24	<b>MasterDataPerr: master data parity error.</b> Read; Write-1-to-clear. 1=ParityErrorEn=1 and either a poisoned completion was received or the device poisoned a write request.
23	<b>FastBackCapable: fast back-to-back capable.</b> Read-only.
22	<b>UDFEn: UDF enable.</b> Read-only.
21	<b>PCI66En: 66 MHz capable.</b> Read-only.
20	<b>CapList: capability list.</b> Read-only. 1=Capability list supported.
19	<b>IntStatus: interrupt status.</b> Read-only. 1=INTx interrupt message pending.
18:11	Reserved.
10	<b>IntDis: interrupt disable.</b> Read-write. 1=INTx interrupt messages generation disabled.
9	<b>FastB2BEn: fast back-to-back enable.</b> Read-only.
8	<b>SerrEn: System error enable.</b> Read-write. 1=Enables reporting of non-fatal and fatal errors detected.
7	<b>Stepping: Stepping control.</b> Read-only.
6	<b>ParityErrorEn: parity error response enable.</b> Read-write.
5	<b>PalSnoopEn: VGA palette snoop enable.</b> Read-only.
4	<b>MemWriteInvalidateEn: memory write and invalidate enable.</b> Read-only.
3	<b>SpecialCycleEn: special cycle enable.</b> Read-only.

2	<b>BusMasterEn: bus master enable.</b> Read-write. 1=Memory and IO read and write request generation enabled.
1	<b>MemAccessEn: IO access enable.</b> Read-write. This bit controls if memory accesses targeting this device are accepted. 1=Enabled. 0=Disabled.
0	<b>IoAccessEn: IO access enable.</b> 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	<b>ClassCode.</b> Value: 03_0000h.
7:0	<b>RevID: revision ID.</b> Value: Product-specific.

### D1F0x0C Header Type Register

Reset: 0080\_0000h.

Bits	Description
31:24	<b>BIST.</b> Read-only.
23:16	<b>HeaderTypeReg.</b> Read-only. The header type field indicates a header type 0 and that this is a multi-function device.
15:8	<b>LatencyTimer.</b> Read-only. These bits are fixed at their default value.
7:0	<b>CacheLineSize.</b> Read-write. This field specifies the system cache line size in units of double words.

### D1F0x10 Graphic Memory Base Address

IF (D0F0xD4\_x0109\_14E2[StrapBifF064BarDisA]==1) THEN Reset: 0000\_0008h. ELSE Reset: 0000\_000Ch. ENDIF.

Bits	Description										
31:26	<b>BaseAddr[31:26]: base address.</b> Read-write. The amount of memory requested by the graphics memory BAR is controlled by D0F0xD4_x0109_1507[StrapBifMemApSizePin] and D0F0xD4_x0109_14E1[StrapBifMemApSize].										
25:4	<b>BaseAddr[25:4]: base address.</b> Read-only.										
3	<b>Pref: prefetchable.</b> Read-only. 1=Prefetchable memory region.										
2:1	<b>Type: base address register type.</b> Read-only. <table border="1" data-bbox="267 1591 665 1766"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>32-bit BAR</td> </tr> <tr> <td>01b</td> <td>Reserved</td> </tr> <tr> <td>10b</td> <td>64-bit BAR</td> </tr> <tr> <td>11b</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Description	00b	32-bit BAR	01b	Reserved	10b	64-bit BAR	11b	Reserved
Bits	Description										
00b	32-bit BAR										
01b	Reserved										
10b	64-bit BAR										
11b	Reserved										
0	<b>MemSpace: memory space type.</b> Read-only. 0=Memory mapped base address.										



**D1F0x14 Graphics Memory Base Address 64**

Reset: 0000\_0000h.

Bits	Description
31:0	<b>BaseAddr[63:32]: base address.</b> Read-write. This field is reserved if (D0F0xD4_x0109_14E2[StrapBifF064BarDisA]==1).

**D1F0x18 Graphics Doorbell Base Address**

IF (D0F0xD4\_x0109\_14E2[StrapBifF064BarDisA]==1) THEN Reset: 0000\_0008h. ELSE Reset: 0000\_000Ch. ENDIF. This register is reserved and reset is 0000\_0000h if (D0F0xD4\_x0109\_14E1[StrapBifDoorbellBarDis]==1).

Bits	Description										
31:23	<b>BaseAddr[31:23]: base address.</b> Read-write.										
22:4	<b>BaseAddr[22:4]: base address.</b> Read-only.										
3	<b>Pref: prefetchable.</b> Read-only. 1=Prefetchable memory region.										
2:1	<b>Type: base address register type.</b> Read-only. <table border="1" data-bbox="267 892 665 1060"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>32-bit BAR</td> </tr> <tr> <td>01b</td> <td>Reserved</td> </tr> <tr> <td>10b</td> <td>64-bit BAR</td> </tr> <tr> <td>11b</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Description	00b	32-bit BAR	01b	Reserved	10b	64-bit BAR	11b	Reserved
Bits	Description										
00b	32-bit BAR										
01b	Reserved										
10b	64-bit BAR										
11b	Reserved										
0	<b>MemSpace: memory space type.</b> Read-only. 0=Memory mapped base address.										

**D1F0x1C Graphics Doorbell Base Address 64**

Reset: 0000\_0000h.

Bits	Description
31:0	<b>BaseAddr[63:32]: base address.</b> Read-write. This field is reserved if (D0F0xD4_x0109_14E1[StrapBifDoorbellBarDis]==1    D0F0xD4_x0109_14E2[StrapBifF064BarDisA]==1).

**D1F0x20 Graphics IO Base Address**

Reset: 0000\_0000h. This register is called Base Address 4 if (D0F0xD4\_x0109\_14E2[StrapBifF064BarDisA]==1).

Bits	Description
31:0	Reserved.

**D1F0x24 Graphics Memory Mapped Registers Base Address**

Reset: 0000\_0000h.

Bits	Description
31:16	<b>BaseAddr[31:16]: base address.</b> Read-write. The amount of memory requested by the graphics memory mapped registers BAR is controlled by <a href="#">D0F0xD4_x0109_14E1</a> [StrapBifRegApSize].
15:4	<b>BaseAddr[15:4]: base address.</b> Read-only.
3	<b>Pref: prefetchable.</b> Read-only. 0=Non-prefetchable memory region.
2:1	<b>Type: base address register type.</b> Read-only. 00b=32-bit BAR.
0	<b>MemSpace: memory space type.</b> Read-only. 0=Memory mapped base address.

**D1F0x2C Subsystem and Subvendor ID Register**Reset: 0000\_0000h. This register can be modified through [D1F0x4C](#)

Bits	Description
31:16	<b>SubsystemID.</b> Read-only.
15:0	<b>SubsystemVendorID.</b> Read-only.

**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	<b>CapPtr: capabilities pointer.</b> Read-only. Pointer to PM capability.

**D1F0x3C Interrupt Line**

Reset: 0000\_01FFh.

Bits	Description
31:16	Reserved.
15:8	<b>InterruptPin: interrupt pin.</b> Read-only. This field identifies the legacy interrupt message the function uses.
7:0	<b>InterruptLine: interrupt line.</b> Read-write. This field contains the interrupt line routing information.

**D1F0x4C Subsystem and Subvendor ID Mirror**

Reset: 0000\_0000h.

Bits	Description
31:16	<b>SubsystemID.</b> Read-write. This field sets the value in the corresponding field in <a href="#">D1F0x2C</a> .
15:0	<b>SubsystemVendorID.</b> Read-write. This field sets the value in the corresponding field in <a href="#">D1F0x2C</a> .

**D1F0x50 Power Management Capability**

Bits	Description
31:27	<b>PmeSupport.</b> Value: 0_0000b. Indicates that there is no PME support.
26	<b>D2Support: D2 support.</b> Value: 1. D2 is supported
25	<b>D1Support: D1 support.</b> Value: 1. D1 is supported
24:22	<b>AuxCurrent: auxiliary current.</b> Value: 0.
21	<b>DevSpecificInit: device specific initialization.</b> Value: 0. Indicates that there is no device specific initialization necessary.
20	Reserved.
19	<b>PmeClock.</b> Value: 0.
18:16	<b>Version: version.</b> Value: 011b.
15:8	<b>NextPtr: next pointer.</b> Value: 58h
7:0	<b>CapID: capability ID.</b> 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	<b>PmeData.</b> Read-only.
23	<b>BusPwrEn.</b> Read-only.
22	<b>B2B3Support.</b> Read-only. B states are not supported.
21:16	Reserved.
15	<b>PmeStatus: PME status.</b> Read-only.
14:13	<b>DataScale: data scale.</b> Read-only.
12:9	<b>DataSelect: data select.</b> Read-only.
8	<b>PmeEn: PME# enable.</b> Read-only.
7:4	Reserved.
3	<b>NoSoftReset: no soft reset.</b> Read-only. Software is required to re-initialize the function when returning from D3hot.

2	Reserved.						
1:0	<b>PowerState: power state.</b> 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. <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>D0</td> </tr> <tr> <td>11b</td> <td>D3hot</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	00b	D0	11b	D3hot
<u>Bits</u>	<u>Definition</u>						
00b	D0						
11b	D3hot						

**D1F0x58 PCI Express® Capability**

Bits	Description
31:30	Reserved.
29:25	<b>IntMessageNum: interrupt message number.</b> Value: 0. This field indicates which MSI vector is used for the interrupt message.
24	<b>SlotImplemented: Slot implemented.</b> Value: 0.
23:20	<b>DeviceType: device type.</b> Value: 9h.
19:16	<b>Version.</b> Value: 2h.
15:8	<b>NextPtr: next pointer.</b> Value: Product-specific.
7:0	<b>CapID: capability ID.</b> Value: 10h.

**D1F0x5C Device Capability**

Bits	Description
31:29	Reserved.
28	<b>FlrCapable: function level reset capability.</b> Value: 0.
27:26	<b>CapturedSlotPowerScale: captured slot power limit scale.</b> Value: 0.
25:18	<b>CapturedSlotPowerLimit: captured slot power limit value.</b> Value: 0.
17:16	Reserved.
15	<b>RoleBasedErrReporting: role-based error reporting.</b> Value: 1.
14:12	Reserved.
11:9	<b>L1AcceptableLatency: endpoint L1 Acceptable Latency.</b> Value: 111b.
8:6	<b>L0SAcceptableLatency: endpoint L0s Acceptable Latency.</b> Value: 110b.
5	<b>ExtendedTag: extended tag support.</b> Value: 1. 8 bit tag support.
4:3	<b>PhantomFunc: phantom function support.</b> Value: 0. No phantom functions supported.
2:0	<b>MaxPayloadSupport: maximum supported payload size.</b> Value: 000b. 128 bytes max payload size.

**D1F0x60 Device Control and Status**

Reset: 0000\_0810h.

Bits	Description
31:22	Reserved.

21	<b>TransactionsPending: transactions pending.</b> Read-only.
20	<b>AuxPwr: auxiliary power.</b> Read-only.
19	<b>UsrDetected: unsupported request detected.</b> Read; Write-1-to-clear. 1=Unsupported request received.
18	<b>FatalErr: fatal error detected.</b> Read; Write-1-to-clear. 1=Fatal error detected.
17	<b>NonFatalErr: non-fatal error detected.</b> Read; Write-1-to-clear. 1=Non-fatal error detected.
16	<b>CorrErr: correctable error detected.</b> Read; Write-1-to-clear. 1=Correctable error detected.
15	<b>BridgeCfgRetryEn: bridge configuration retry enable.</b> Read-only.
14:12	<b>MaxRequestSize: maximum request size.</b> Read-only.
11	<b>NoSnoopEnable: enable no snoop.</b> Read-write. 1=The device is permitted to set the No Snoop bit in requests.
10	<b>AuxPowerPmEn: auxiliary power PM enable.</b> Read-only. This capability is not implemented.
9	<b>PhantomFuncEn: phantom functions enable.</b> Read-only. Phantom functions are not supported.
8	<b>ExtendedTagEn: extended tag enable.</b> Read-write. 1=8-bit tag request tags. 0=5-bit request tag.
7:5	<b>MaxPayloadSize: maximum supported payload size.</b> Read-only. 000b=Indicates a 128 byte maximum payload size.
4	<b>RelaxedOrdEn: relaxed ordering enable.</b> Read-write. 1=The device is permitted to set the Relaxed Ordering bit.
3	<b>UsrReportEn: unsupported request reporting enable.</b> Read-write. 1=Enables signaling unsupported requests by sending error messages.
2	<b>FatalErrEn: fatal error reporting enable.</b> Read-write. 1=Enables sending ERR_FATAL message when a fatal error is detected.
1	<b>NonFatalErrEn: non-fatal error reporting enable.</b> Read-write. 1=Enables sending ERR_NONFATAL message when a non-fatal error is detected.
0	<b>CorrErrEn: correctable error reporting enable.</b> Read-write. 1=Enables sending ERR_CORR message when a correctable error is detected.

### D1F0x64 Link Capability

Bits	Description
31:24	<b>PortNumber: port number.</b> Read-only. Value: 0. This field indicates the PCI Express® port number for the given PCI Express® link.
23	Reserved.
22	<b>AspmOptionalityCompliance: ASP Optionality ECN capability.</b> Read-only. Value: 0b.
21	<b>LinkBWNotificationCap: link bandwidth notification capability.</b> Read-only. Value: 0b.
20	<b>DLActiveReportingCapable: data link layer active reporting capability.</b> Read-only. Value: 0b.
19	<b>SurpriseDownErrReporting: surprise down error reporting capability.</b> Read-only. Value: 0b.
18	<b>ClockPowerManagement: clock power management.</b> Read-only. Value: 0b.
17:15	<b>L1ExitLatency: L1 exit latency.</b> Read-only. Value: 0b.
14:12	<b>L0sExitLatency: L0s exit latency.</b> Read-only. Value: 0b.
11:10	<b>PMSupport: active state power management support.</b> Read-only. Value: 0b.

9:4	<b>LinkWidth: maximum link width.</b> Read-only. Value: 0.
3:0	<b>LinkSpeed: link speed.</b> Read-only. Value: 0b.

### D1F0x68 Link Control and Status

Reset: 0000\_0000h.

Bits	Description												
31	<b>LinkAutonomousBWStatus: link autonomous bandwidth status.</b> Read-only.												
30	<b>LinkBWManagementStatus: link bandwidth management status.</b> Read-only.												
29	<b>DLActive: data link layer link active.</b> 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	<b>SlotClockCfg: slot clock configuration.</b> Read-only. 1=the root port uses the same clock that the platform provides.												
27	<b>LinkTraining: link training.</b> 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	<b>NegotiatedLinkWidth: negotiated link width.</b> Read-only. This field indicates the negotiated width of the given PCI Express® link.												
19:16	<b>LinkSpeed: link speed.</b> Read-only.												
15:12	Reserved.												
11	<b>LinkAutonomousBWIntEn: link autonomous bandwidth interrupt enable.</b> Read-only.												
10	<b>LinkBWManagementEn: link bandwidth management interrupt enable.</b> Read-only.												
9	<b>HWAutonomousWidthDisable: hardware autonomous width disable.</b> Read-only. 1=Hardware not allowed to change the link width except to correct unreliable link operation by reducing link width.												
8	<b>ClockPowerManagementEn: clock power management enable.</b> Read-only.												
7	<b>ExtendedSync: extended sync.</b> Read-only. 1=Forces the transmission of additional ordered sets when exiting the L0s state and when in the recovery state.												
6	<b>CommonClockCfg: common clock configuration.</b> Read-only. 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	<b>RetrainLink: retrain link.</b> Read-only. This bit does not apply to endpoints.												
4	<b>LinkDis: link disable.</b> Read-only. This bit does not apply to endpoints.												
3	<b>ReadCplBoundary: read completion boundary.</b> Read-only. 0=64 byte read completion boundary.												
2	Reserved.												
1:0	<b>PmControl: active state power management enable.</b> Read-only. This field controls the level of ASPM supported on the given PCI Express® link. <table border="1" data-bbox="267 1837 1372 1938"> <thead> <tr> <th>Bits</th> <th>Definition</th> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Disabled.</td> <td>10b</td> <td>L1 Entry Enabled.</td> </tr> <tr> <td>01b</td> <td>L0s Entry Enabled.</td> <td>11b</td> <td>L0s and L1 Entry Enabled.</td> </tr> </tbody> </table>	Bits	Definition	Bits	Definition	00b	Disabled.	10b	L1 Entry Enabled.	01b	L0s Entry Enabled.	11b	L0s and L1 Entry Enabled.
Bits	Definition	Bits	Definition										
00b	Disabled.	10b	L1 Entry Enabled.										
01b	L0s Entry Enabled.	11b	L0s and L1 Entry Enabled.										

**D1F0x7C Device Capability 2**

Reset: 0000\_0000h.

Bits	Description												
31:24	Reserved.												
23:22	<b>MaxEndEndTlpPrefixes: Max number of End-End TLP prefixes supported.</b> Read-only. IF (D1F0x7C[EndEndTlpPrefixSupported]==0) THEN Reserved. ENDIF. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>4 End-End TLP Prefixes.</td> <td>10b</td> <td>2 End-End TLP Prefixes.</td> </tr> <tr> <td>01b</td> <td>1 End-End TLP Prefix.</td> <td>11b</td> <td>3 End-End TLP Prefixes.</td> </tr> </tbody> </table>	Bits	Definition	Bits	Definition	00b	4 End-End TLP Prefixes.	10b	2 End-End TLP Prefixes.	01b	1 End-End TLP Prefix.	11b	3 End-End TLP Prefixes.
Bits	Definition	Bits	Definition										
00b	4 End-End TLP Prefixes.	10b	2 End-End TLP Prefixes.										
01b	1 End-End TLP Prefix.	11b	3 End-End TLP Prefixes.										
21	<b>EndEndTlpPrefixSupported: End-End TLP Prefix supported.</b> Read-only.												
20	<b>ExtendedFmtFieldSupported.</b> Read-only. 1=Function supports 3-bit definition of Fmt field. 0=Function supports 2-bit definition of Fmt field.												
19:18	<b>ObffSupported: Optimized buffer flush/fill supported.</b> Read-only.												
17:14	Reserved.												
13:12	<b>TphCplrSupported.</b> Read-only.												
11	<b>LtrSupported: Latency Tolerance Reporting supported.</b> Read-only.												
10	<b>NoRoEnabledP2pPassing.</b> Read-only.												
9:6	Reserved.												
5	<b>AriForwardingSupported: ARI forwarding supported.</b> Read-only.												
4	<b>CplTimeoutDisSupported: completion timeout disable supported.</b> Read-only.												
3:0	<b>CplTimeoutRangeSupported: completion timeout range supported.</b> Read-only.												

**D1F0x80 Device Control and Status 2**

Reset: 0000\_0000h.

Bits	Description
31:16	Reserved.
15	<b>EndEndTlpPrefixBlocking.</b> Read-only.
14:13	<b>ObffEn.</b> Read-only.
12:11	Reserved.
10	<b>LtrEn.</b> Read-only.
9	<b>IdoCompletionEn.</b> Read-only.
8	<b>IdoRequestEn.</b> Read-only.
7:6	Reserved.
5	<b>AriForwardingEn.</b> Read-only.
4	<b>CplTimeoutDis: completion timeout disable.</b> Read-only.
3:0	<b>CplTimeoutValue: completion timeout range supported.</b> Read-only.

**D1F0x84 Link Capability 2**

Bits	Description
31:9	Reserved.
8	<b>CrosslinkSupported: Crosslink Spported.</b> Read-only. Reset: 0. 1=Crosslink supported.
7:1	<b>SupportedLinkSpeed: Supported Link Speed.</b> Read-only. Reset: 07h. Specifies what link speeds are supported. Bit 1 = 2.5 GT/s, Bit2 = 5.0 Gt/s, Bit 3 = 8.0 GT/s.
0	Reserved.

**D1F0x88 Link Control and Status 2**

Reset: 0000\_0000h.

Bits	Description
31:22	Reserved.
21	<b>LinkEqualizationRequest.</b> Read-only. Set when hardware requests link equalization to be performed.
20	<b>EqualizationPhase3Success: Phase3 of Tx equalization procedure completed.</b> Read-only.
19	<b>EqualizationPhase2Success: Phase2 of Tx equalization procedure completed.</b> Read-only.
18	<b>EqualizationPhase1Success: Phase1 of Tx equalization procedure completed.</b> Read-only.
17	<b>EqualizationComplete: Tx equalization procedure completed.</b> Read-only.
16	<b>CurDeemphasisLevel: current deemphasis level.</b> Read-only. 1=-3.5 dB. 0=-6 dB.
15:13	Reserved.
12	<b>ComplianceDeemphasis: compliance deemphasis.</b> Read-only. This bit defines the deemphasis level used in compliance mode. 1=-3.5 dB. 0=-6 dB.
11	<b>ComplianceSOS: compliance SOS.</b> Read-only. 1=The device transmits skip ordered sets in between the modified compliance pattern.
10	<b>EnterModCompliance: enter modified compliance.</b> Read-only. 1=The device transmits modified compliance pattern.
9:7	<b>XmitMargin: transmit margin.</b> Read-only. This field controls the non-deemphasized voltage level at the transmitter pins.
6	<b>SelectableDeemphasis: selectable deemphasis.</b> Read-only.
5	<b>HwAutonomousSpeedDisable: hardware autonomous speed disable.</b> Read-only. 1=Disables hardware generated link speed changes.
4	<b>EnterCompliance: enter compliance.</b> Read-only. 1=Force link to enter compliance mode.
3:0	<b>TargetLinkSpeed: target link speed.</b> Read-only. This field defines the upper limit of the link operational speed.

**D1F0xA0 MSI Capability**

Bits	Description
31:24	Reserved.



23	<b>Msi64bit: MSI 64 bit capability.</b> Read-only. Reset: 1. 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	<b>MsiMultiEn: MSI multiple message enable.</b> 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	<b>MsiMultiCap: MSI multiple message capability.</b> Read-only. Reset: 000b. 000b=The device is requesting one vector.
16	<b>MsiEn: MSI enable.</b> 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	<b>NextPtr: next pointer.</b> Read-only. Reset:00h
7:0	<b>CapID: capability ID.</b> Read-only. Reset: 05h. 05h=MSI capability structure.

### D1F0xA4 MSI Message Address Low

Reset: 0000\_0000h.

Bits	Description
31:2	<b>MsiMsgAddrLo: MSI message address.</b> Read-write. This register specifies the dword aligned address for the MSI memory write transaction.
1:0	Reserved.

### D1F0xA8 MSI Message Address High

Reset: 0000\_0000h.

Bits	Description
31:8	Reserved.
7:0	<b>MsiMsgAddrHi: MSI message address.</b> Read-write. This register specifies the upper 8-bits of the MSI address in 64 bit MSI mode.

### D1F0xAC MSI Message Data

Reset: 0000\_0000h.

Bits	Description
31:16	Reserved.
15:0	<b>MsiData: MSI message data.</b> 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: 0061\_000Bh.

Bits	Description
31:20	<b>NextPtr: next pointer.</b> Read-only.

19:16	<b>CapVer: capability version.</b> Read-only.
15:0	<b>CapID: capability ID.</b> Read-only.

### D1F0x104 Vendor Specific Header

---

Reset: 0101\_0001h.

Bits	Description
31:20	<b>VsecLen: vendor specific enhanced next pointer.</b> Read-only.
19:16	<b>VsecRev: vendor specific enhanced capability version.</b> Read-only.
15:0	<b>VsecID: vendor specific enhanced capability ID.</b> Read-only.

### D1F0x108 Vendor Specific 1

---

Reset: 0000\_0000h.

Bits	Description
31:0	<b>Scratch: scratch.</b> Read-write.

### D1F0x10C Vendor Specific 2

---

Reset: 0000\_0000h.

Bits	Description
31:0	<b>Scratch: scratch.</b> Read-write.

### 3.6 Device 1 Function 1 (Audio Controller) 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.

#### D1F1x00 Device/Vendor ID

Bits	Description
31:16	<b>DeviceID: device ID.</b> Read-only. Value: Product-specific. .
15:0	<b>VendorID: vendor ID.</b> Read-only. Value: 1002h.

#### D1F1x04 Status/Command

Reset: 0010\_0000h.

Bits	Description
31	<b>ParityErrorDetected: detected parity error.</b> Read; Write-1-to-clear. 1=Poisoned TLP received.
30	<b>SignaledSystemError: signaled system error.</b> Read; Write-1-to-clear. 1=A non-fatal or fatal error message was sent and SerrEn=1.
29	<b>ReceivedMasterAbort: received master abort.</b> Read; Write-1-to-clear. 1=A completion with an unsupported request completion status was received.
28	<b>ReceivedTargetAbort: received target abort.</b> Read; Write-1-to-clear. 1=A completion with completer abort completion status was received.
27	<b>SignalTargetAbort: Signaled target abort.</b> Read-only.
26:25	<b>DevselTiming: DEVSEL# Timing.</b> Read-only.
24	<b>MasterDataPerr: master data parity error.</b> Read; Write-1-to-clear. 1=ParityErrorEn=1 and either a poisoned completion was received or the device poisoned a write request.
23	<b>FastBackCapable: fast back-to-back capable.</b> Read-only.
22	<b>UDFEn: UDF enable.</b> Read-only.
21	<b>PCI66En: 66 MHz capable.</b> Read-only.
20	<b>CapList: capability list.</b> Read-only. 1=capability list supported.
19	<b>IntStatus: interrupt status.</b> Read-only. 1=INTx interrupt message pending.
18:11	Reserved.
10	<b>IntDis: interrupt disable.</b> Read-write. 1=INTx interrupt messages generation disabled.
9	<b>FastB2BEn: fast back-to-back enable.</b> Read-only.
8	<b>SerrEn: System error enable.</b> Read-write. 1=Enables reporting of non-fatal and fatal errors detected.
7	<b>Stepping: Stepping control.</b> Read-only.
6	<b>ParityErrorEn: parity error response enable.</b> Read-write.
5	<b>PalSnoopEn: VGA palette snoop enable.</b> Read-only.

4	<b>MemWriteInvalidateEn: memory write and invalidate enable.</b> Read-only.
3	<b>SpecialCycleEn: special cycle enable.</b> Read-only.
2	<b>BusMasterEn: bus master enable.</b> Read-write. 1=Memory and IO read and write request generation enabled.
1	<b>MemAccessEn: IO access enable.</b> Read-write. This bit controls if memory accesses targeting this device are accepted. 1=Enabled. 0=Disabled.
0	<b>IoAccessEn: IO access enable.</b> 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	<b>ClassCode.</b> Read-only.
7:0	<b>RevID: revision ID.</b> Read-only.

**D1F1x0C Header Type**

Reset: 0080\_0000h.

Bits	Description
31:24	<b>BIST.</b> Read-only. These bits are fixed at their default values.
23:16	<b>HeaderTypeReg.</b> Read-only. 80h=Type 0 multi-function device.
15:8	<b>LatencyTimer.</b> Read-only. These bits are fixed at their default value.
7:0	<b>CacheLineSize.</b> 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	<b>BaseAddr: base address.</b> Read-write.
13:4	Reserved.
3	<b>Pref: prefetchable.</b> Read-only. 0=Non-prefetchable memory region.
2:1	<b>Type: base address register type.</b> Read-only. 00b=32-bit base address register.
0	<b>MemSpace: memory space type.</b> 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**Reset: 0000\_0000h. This register can be modified through [D1F1x4C](#).

Bits	Description
31:16	<b>SubsystemID</b> . Read-only.
15:0	<b>SubsystemVendorID</b> . Read-only.

**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	<b>CapPtr: capabilities pointer.</b> Read-only. Pointer to PM capability.

**D1F1x3C Interrupt Line**

Reset: 0000\_02FFh.

Bits	Description
31:16	Reserved.
15:8	<b>InterruptPin: interrupt pin.</b> Read-only. This field identifies the legacy interrupt message the function uses.
7:0	<b>InterruptLine: interrupt line.</b> Read-write. This field contains the interrupt line routing information.

**D1F1x4C Subsystem and Subvendor ID Mirror**

Reset: 0000\_0000h.

Bits	Description
31:16	<b>SubsystemID.</b> Read-write. This field sets the value in the corresponding field in <a href="#">D1F1x2C</a> .
15:0	<b>SubsystemVendorID.</b> Read-write. This field sets the value in the corresponding field in <a href="#">D1F1x2C</a> .

**D1F1x50 Power Management Capability**

Bits	Description
31:27	<b>PmeSupport.</b> Value: 0_0000b. Indicates that there is no PME support.
26	<b>D2Support: D2 support.</b> Value: 1. D2 is supported
25	<b>D1Support: D1 support.</b> Value: 1. D1 is supported
24:22	<b>AuxCurrent: auxiliary current.</b> Value: 0.
21	<b>DevSpecificInit: device specific initialization.</b> Value: 0. Indicates that there is no device specific initialization necessary.
20	Reserved.
19	<b>PmeClock.</b> Value: 0.
18:16	<b>Version: version.</b> Value: 011b.
15:8	<b>NextPtr: next pointer.</b> Value: 00h.
7:0	<b>CapID: capability ID.</b> 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	<b>PmeData</b> : Read-only.								
23	<b>BusPwrEn</b> : Read-only.								
22	<b>B2B3Support</b> : Read-only. B states are not supported.								
21:16	Reserved.								
15	<b>PmeStatus</b> : <b>PME status</b> . Read-only.								
14:13	<b>DataScale</b> : <b>data scale</b> . Read-only.								
12:9	<b>DataSelect</b> : <b>data select</b> . Read-only.								
8	<b>PmeEn</b> : <b>PME# enable</b> . Read-only.								
7:4	Reserved.								
3	<b>NoSoftReset</b> : <b>no soft reset</b> . Read-only. Software is required to re-initialize the function when returning from D3 <sub>hot</sub> .								
2	Reserved.								
1:0	<b>PowerState</b> : <b>power state</b> . 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. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>D0</td> </tr> <tr> <td>01b, 10b</td> <td>Reserved</td> </tr> <tr> <td>11b</td> <td>D3hot</td> </tr> </tbody> </table>	Bits	Definition	00b	D0	01b, 10b	Reserved	11b	D3hot
Bits	Definition								
00b	D0								
01b, 10b	Reserved								
11b	D3hot								

**D1F1x58 PCI Express® Capability**

Bits	Description
31:30	Reserved.
29:25	<b>IntMessageNum</b> : <b>interrupt message number</b> . Value: 0. This field indicates which MSI vector is used for the interrupt message.
24	<b>SlotImplemented</b> : <b>Slot implemented</b> . Value: 0.
23:20	<b>DeviceType</b> : <b>device type</b> . Value: 9h.
19:16	<b>Version</b> . Value: 2h.
15:8	<b>NextPtr</b> : <b>next pointer</b> . Value: Product-specific.
7:0	<b>CapID</b> : <b>capability ID</b> . Value: 10h.

**D1F1x5C Device Capability**

Bits	Description
31:29	Reserved.
28	<b>FlrCapable</b> : <b>function level reset capability</b> . Value: Product-specific.
27:26	<b>CapturedSlotPowerScale</b> : <b>captured slot power limit scale</b> . Value: 0.

25:18	<b>CapturedSlotPowerLimit:</b> captured slot power limit value. Value: 0.
17:16	Reserved.
15	<b>RoleBasedErrReporting:</b> role-based error reporting. Value: 1.
14:12	Reserved.
11:9	<b>L1AcceptableLatency:</b> endpoint L1 Acceptable Latency. Value: 111b.
8:6	<b>L0SAcceptableLatency:</b> endpoint L0s Acceptable Latency. Value: 110b.
5	<b>ExtendedTag:</b> extended tag support. Value: 1. 8 bit tag support.
4:3	<b>PhantomFunc:</b> phantom function support. Value: 0. No phantom functions supported.
2:0	<b>MaxPayloadSupport:</b> 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	<b>TransactionsPending:</b> transactions pending. Read-only.
20	<b>AuxPwr:</b> auxiliary power. Read-only.
19	<b>UsrDetected:</b> unsupported request detected. Read; Write-1-to-clear. 1=Unsupported request received.
18	<b>FatalErr:</b> fatal error detected. Read; Write-1-to-clear. 1=Fatal error detected.
17	<b>NonFatalErr:</b> non-fatal error detected. Read; Write-1-to-clear. 1=Non-fatal error detected.
16	<b>CorrErr:</b> correctable error detected. Read; Write-1-to-clear. 1=Correctable error detected.
15	<b>BridgeCfgRetryEn:</b> bridge configuration retry enable. Read-only.
14:12	<b>MaxRequestSize:</b> maximum request size. Read-only. 0=The root port never generates read requests with size exceeding 128 bytes.
11	<b>NoSnoopEnable:</b> enable no snoop. Read-write. 1=The device is permitted to set the No Snoop bit in requests.
10	<b>AuxPowerPmEn:</b> auxiliary power PM enable. Read-only. This capability is not implemented.
9	<b>PhantomFuncEn:</b> phantom functions enable. Read-only. Phantom functions are not supported.
8	<b>ExtendedTagEn:</b> extended tag enable. Read-write. 1=8-bit tag request tags. 0=5-bit request tag.
7:5	<b>MaxPayloadSize:</b> maximum supported payload size. Read-only. 000b=Indicates a 128 byte maximum payload size.
4	<b>RelaxedOrdEn:</b> relaxed ordering enable. Read-write. 1=The device is permitted to set the Relaxed Ordering bit.
3	<b>UsrReportEn:</b> unsupported request reporting enable. Read-write. 1=Enables signaling unsupported requests by sending error messages.
2	<b>FatalErrEn:</b> fatal error reporting enable. Read-write. 1=Enables sending ERR_FATAL message when a fatal error is detected.



1	<b>NonFatalErrEn: non-fatal error reporting enable.</b> Read-write. 1=Enables sending ERR_NONFATAL message when a non-fatal error is detected.
0	<b>CorrErrEn: correctable error reporting enable.</b> Read-write. 1=Enables sending ERR_CORR message when a correctable error is detected.

### D1F1x64 Link Capability

Bits	Description
31:24	<b>PortNumber: port number.</b> Value: 0. This field indicates the PCI Express® port number for the given PCI Express® link.
23:22	Reserved.
21	<b>LinkBWNotificationCap: link bandwidth notification capability.</b> Read-only. Value: 0b.
20	<b>DLActiveReportingCapable: data link layer active reporting capability.</b> Read-only. Value: 0b.
19	<b>SurpriseDownErrReporting: surprise down error reporting capability.</b> Read-only. Value: 0b.
18	<b>ClockPowerManagement: clock power management.</b> Read-only. Value: 0b.
17:15	<b>L1ExitLatency: L1 exit latency.</b> Read-only. Value: 0b.
14:12	<b>L0sExitLatency: L0s exit latency.</b> Read-only. Value: 0b.
11:10	<b>PMSupport: active state power management support.</b> Read-only. Value: 0b.
9:4	<b>LinkWidth: maximum link width.</b> Read-only. Value: 0.
3:0	<b>LinkSpeed: link speed.</b> Read-only. Value: 0b.

### D1F1x68 Link Control and Status

Reset: 0000\_0000h.

Bits	Description
31	<b>LinkAutonomousBWStatus: link autonomous bandwidth status.</b> Read-only.
30	<b>LinkBWManagementStatus: link bandwidth management status.</b> Read-only.
29	<b>DLActive: data link layer link active.</b> 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	<b>SlotClockCfg: slot clock configuration.</b> Read-only. 1=The root port uses the same clock that the platform provides.
27	<b>LinkTraining: link training.</b> 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	<b>NegotiatedLinkWidth: negotiated link width.</b> Read-only. This field indicates the negotiated width of the given PCI Express® link.

19:16	<b>LinkSpeed: link speed.</b> Read-only. <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Reserved</td> </tr> <tr> <td>1h</td> <td>2.5 Gb/s.</td> </tr> <tr> <td>2h</td> <td>5 Gb/s.</td> </tr> <tr> <td>Fh-3h</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Description	0h	Reserved	1h	2.5 Gb/s.	2h	5 Gb/s.	Fh-3h	Reserved		
Bits	Description												
0h	Reserved												
1h	2.5 Gb/s.												
2h	5 Gb/s.												
Fh-3h	Reserved												
15:12	Reserved.												
11	<b>LinkAutonomousBWIntEn: link autonomous bandwidth interrupt enable.</b> Read-only.												
10	<b>LinkBWManagementEn: link bandwidth management interrupt enable.</b> Read-only.												
9	<b>HWAutonomousWidthDisable: hardware autonomous width disable.</b> Read-only. 1=Hardware not allowed to change the link width except to correct unreliable link operation by reducing link width.												
8	<b>ClockPowerManagementEn: clock power management enable.</b> Read-only.												
7	<b>ExtendedSync: extended sync.</b> Read-only. 1=Forces the transmission of additional ordered sets when exiting the L0s state and when in the recovery state.												
6	<b>CommonClockCfg: common clock configuration.</b> Read-only. 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	<b>RetrainLink: retrain link.</b> Read-only. This bit does not apply to endpoints.												
4	<b>LinkDis: link disable.</b> Read-only. This bit does not apply to endpoints.												
3	<b>ReadCplBoundary: read completion boundary.</b> Read-only. 0=64 byte read completion boundary.												
2	Reserved.												
1:0	<b>PmControl: active state power management enable.</b> Read-only. This field controls the level of ASPM supported on the given PCI Express® link. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Disabled.</td> <td>10b</td> <td>L1 Entry Enabled.</td> </tr> <tr> <td>01b</td> <td>L0s Entry Enabled.</td> <td>11b</td> <td>L0s and L1 Entry Enabled.</td> </tr> </tbody> </table>	Bits	Definition	Bits	Definition	00b	Disabled.	10b	L1 Entry Enabled.	01b	L0s Entry Enabled.	11b	L0s and L1 Entry Enabled.
Bits	Definition	Bits	Definition										
00b	Disabled.	10b	L1 Entry Enabled.										
01b	L0s Entry Enabled.	11b	L0s and L1 Entry Enabled.										

**D1F1x7C Device Capability 2**

Reset: 0000\_0000h.

Bits	Description
31:5	Reserved.
4	<b>CplTimeoutDisSup: completion timeout disable supported.</b> Read-only.
3:0	<b>CplTimeoutRangeSup: completion timeout range supported.</b> Read-only.

**D1F1x80 Device Control and Status 2**

Reset: 0000\_0000h.

Bits	Description
31:5	Reserved.
4	<b>CplTimeoutDis: completion timeout disable.</b> Read-only.
3:0	<b>CplTimeoutValue: completion timeout range supported.</b> 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	<b>CurDeemphasisLevel: current deemphasis level.</b> Read-only. 1=-3.5 dB. 0=-6 dB.
15:13	Reserved.
12	<b>ComplianceDeemphasis: compliance deemphasis.</b> Read-only. This bit defines the deemphasis level used in compliance mode. 1=-3.5 dB. 0=-6 dB.
11	<b>ComplianceSOS: compliance SOS.</b> Read-only. 1=The device transmits skip ordered sets in between the modified compliance pattern.
10	<b>EnterModCompliance: enter modified compliance.</b> Read-only. 1=The device transmits modified compliance pattern.
9:7	<b>XmitMargin: transmit margin.</b> Read-only. This field controls the non-deemphasized voltage level at the transmitter pins.
6	<b>SelectableDeemphasis: selectable deemphasis.</b> Read-only.
5	<b>HwAutonomousSpeedDisable: hardware autonomous speed disable.</b> Read-only. 1=Disables hardware generated link speed changes.
4	<b>EnterCompliance: enter compliance.</b> Read-only. 1=Force link to enter compliance mode.
3:0	<b>TargetLinkSpeed: target link speed.</b> Read-only. This fields defines the upper limit of the link operational speed.

**D1F1xA0 MSI Capability**

Bits	Description
31:24	Reserved.
23	<b>Msi64bit: MSI 64 bit capability.</b> Read-only. Reset: 1. 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	<b>MsiMultiEn: MSI multiple message enable.</b> 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	<b>MsiMultiCap: MSI multiple message capability.</b> Read-only. Reset:000b. 000b=The device is requesting one vector.
16	<b>MsiEn: MSI enable.</b> 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	<b>NextPtr: next pointer.</b> Read-only. Reset: 00h.
7:0	<b>CapID: capability ID.</b> Read-only. Reset: 05h. 05h=MSI capability structure.

**D1F1xA4 MSI Message Address Low**

Reset: 0000\_0000h.

Bits	Description
31:2	<b>MsiMsgAddrLo: MSI message address.</b> Read-write. This register specifies the dword aligned address for the MSI memory write transaction.
1:0	Reserved.

**D1F1xA8 MSI Message Address High**

Reset: 0000\_0000h.

Bits	Description
31:8	Reserved.
7:0	<b>MsiMsgAddrHi: MSI message address.</b> Read-write. This register specifies the upper 8-bits of the MSI address in 64 bit MSI mode.

**D1F1xAC MSI Message Data**

Reset: 0000\_0000h.

Bits	Description
31:16	Reserved.
15:0	<b>MsiData: MSI message data.</b> 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: 0111\_000Bh.

Bits	Description
31:20	<b>NextPtr: next pointer.</b> Read-only.
19:16	<b>CapVer: capability version.</b> Read-only.
15:0	<b>CapID: capability ID.</b> Read-only.

**D1F1x104 Vendor Specific Header**

Reset: 0101\_0001h.

Bits	Description
31:20	<b>VsecLen: vendor specific enhanced next pointer.</b> Read-only.
19:16	<b>VsecRev: vendor specific enhanced capability version.</b> Read-only.
15:0	<b>VsecID: vendor specific enhanced capability ID.</b> Read-only.

**D1F1x108 Vendor Specific 1**

---

Reset: 0000\_0000h.

Bits	Description
31:0	<b>Scratch: scratch.</b> Read-write.

**D1F1x10C Vendor Specific 2**

---

Reset: 0000\_0000h.

Bits	Description
31:0	<b>Scratch: scratch.</b> Read-write.

### 3.7 Device 2 Function 0 (Host Bridge) 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. D2F0 registers do not control any hardware. They ensure that software can configure functions 1 through 4.

#### D2F0x00 Device/Vendor ID (Host Bridge)

Bits	Description
31:16	<b>DeviceID: device ID.</b> Read-only. Value: 1538h.
15:0	<b>VendorID: vendor ID.</b> Read-only. Value: 1022h.

#### D2F0x04 Status/Command

Reset: 0000\_0000h.

Bits	Description
31:16	<b>Status.</b> Read-only.
15:0	<b>Command.</b> Read-only.

#### D2F0x08 Class Code/Revision ID

Reset: 0600\_0000h.

Bits	Description
31:8	<b>ClassCode: class code.</b> Read-only.
7:0	<b>RevId: revision identifier.</b> Read-only.

#### D2F0x0C Header Type

Reset: 0080\_0000h.

Bits	Description
31:24	Reserved.
23	<b>DeviceType.</b> Read-only. 1=Indicates that the northbridge block is a multi-function device. 0=Indicates that the northbridge block is a single function device.
22:16	<b>HeaderType.</b> Read-only. Indicates multiple functions present in this device.
15:0	Reserved.

**D2F0x40 Header Type Write**

Reset: 0000\_0080h.

Bits	Description
31:8	Reserved.
7	<b>DeviceType</b> . Read-write. This field sets the value in <a href="#">D2F0x0C[DeviceType]</a> . 0=Single function device. 1=Multi-function device.
6:0	Reserved.

**3.8 Device 2 Function [5:1] (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. See 2.11.1 [Overview].

**D2F[5:1]x00 Device/Vendor ID**

Table 99: Register Mapping for D2F[5:1]x00

D2F[5:1]x00	Function
D2F1x00	GPP Bridge 0
D2F2x00	GPP Bridge 1
D2F3x00	GPP Bridge 2
D2F4x00	GPP Bridge 3
D2F5x00	GPP Bridge 4

Bits	Description
31:16	<b>DeviceID: device ID</b> . Read-only. Value: 1439h.
15:0	<b>VendorID: vendor ID</b> . Read-only. Value: 1022h.

**D2F[5:1]x04 Status/Command Register**

Reset: 0010\_0000h.

Bits	Description
31	<b>ParityErrorDetected: detected parity error</b> . Read; Write-1-to-clear.
30	<b>SignaledSystemError: signaled system error</b> . Read; Write-1-to-clear. 1=System error signalled.
29	<b>ReceivedMasterAbort: received master abort</b> . Read; Write-1-to-clear.
28	<b>ReceivedTargetAbort: received target abort</b> . Read; Write-1-to-clear.
27	<b>SignalTargetAbort: signaled target abort</b> . Read; Write-1-to-clear.
26:25	<b>DevSelTiming: DEVSEL# Timing</b> . Read-only.

24	<b>DataPerr: data parity error.</b> Read; Write-1-to-clear.
23	<b>FastBackCapable: fast back-to-back capable.</b> Read-only.
22	Reserved.
21	<b>PCI66En: 66 MHz capable.</b> Read-only.
20	<b>CapList: capability list.</b> Read-only. 1= Capability list present.
19	<b>IntStatus: interrupt status.</b> Read-only. 1=An INTx interrupt Message is pending in the device.
18:11	Reserved.
10	<b>IntDis: interrupt disable.</b> Read-write.
9	<b>FastB2BEn: fast back-to-back enable.</b> Read-only.
8	<b>SerrEn: system error enable.</b> Read-write. 1=System error reporting enabled.
7	<b>Stepping: Stepping control.</b> Read-only.
6	<b>ParityErrorEn: parity error response enable.</b> Read-write.
5	<b>PalSnoopEn: VGA palette snoop enable.</b> Read-only.
4	<b>MemWriteInvalidateEn: memory write and invalidate enable.</b> Read-only.
3	<b>SpecialCycleEn: special cycle enable.</b> Read-only.
2	<b>BusMasterEn: bus master enable.</b> Read-write.
1	<b>MemAccessEn: IO access enable.</b> Read-write. This bit controls if memory accesses targeting this device are accepted or not. 1=Enabled. 0=Disabled.
0	<b>IoAccessEn: IO access enable.</b> Read-write. This bit controls if IO accesses targeting this device are accepted or not. 1=Enabled. 0=Disabled.

### D2F[5:1]x08 Class Code/Revision ID Register

Reset: 0604\_00xxh.

Bits	Description
31:8	<b>ClassCode.</b> Read-only. Provides the host bridge class code as defined in the PCI specification.
7:0	<b>RevID: revision ID.</b> Read-only.

### D2F[5:1]x0C Header Type Register

Reset: 0001\_0000h.

Bits	Description
31:24	<b>BIST.</b> Read-only. These bits are fixed at their default values.
23	<b>DeviceType.</b> Read-only. 0=Single function device. 1=Multi-function device.
22:16	<b>HeaderType.</b> Read-only. These bits are fixed at their default values. Indicates a Type 0 or Type 1 configuration space.
15:8	<b>LatencyTimer.</b> Read-only. This field does not control any hardware.
7:0	<b>CacheLineSize.</b> Read-write.



**D2F[5:1]x18 Bus Number and Secondary Latency Register**

Reset: 0000\_0000h.

Bits	Description
31:24	<b>SecondaryLatencyTimer: secondary latency timer.</b> Read-only. This field is always 0.
23:16	<b>SubBusNumber: subordinate number.</b> Read-write. This field contains the highest-numbered bus that exists on the secondary side of the bridge.
15:8	<b>SecondaryBus: secondary bus number.</b> Read-write. This field defines the bus number of the secondary bus interface.
7:0	<b>PrimaryBus: primary bus number.</b> Read-write. This field defines the bus number of the primary bus interface.

**D2F[5:1]x1C IO Base and Secondary Status Register**

Reset: 0000\_0101h.

Bits	Description
31	<b>ParityErrorDetected: detected parity error.</b> Read; Write-1-to-clear. A Poisoned TLP was received regardless of the state of the <a href="#">D2F[5:1]x04[ParityErrorEn]</a> .
30	<b>ReceivedSystemError: signaled system error.</b> Read; Write-1-to-clear. 1=A System Error was detected.
29	<b>ReceivedMasterAbort: received master abort.</b> Read; Write-1-to-clear. 1=A CPU transaction is terminated due to a master-abort.
28	<b>ReceivedTargetAbort: received target abort.</b> Read; Write-1-to-clear. 1=A CPU transaction (except for a special cycle) is terminated due to a target-abort.
27	<b>SignalTargetAbort: signaled target abort.</b> Read; Write-1-to-clear.
26:25	<b>DevselTiming: DEVSEL# Timing.</b> Read-only.
24	<b>MasterDataPerr: master data parity error.</b> Read; Write-1-to-clear. 1=The link received a poisoned or poisoned a downstream write and <a href="#">D2F[5:1]x3C[ParityResponseEn]=1</a> .
23	<b>FastBackCapable: fast back-to-back capable.</b> Read-only.
22	Reserved.
21	<b>PCI66En: 66 MHz capable.</b> Read-only.
20	<b>CapList: capability list.</b> Read-only.
19:16	Reserved.
15:12	<b>IOLimit[15:12].</b> Read-write. Lower part of the limit address. Upper part is defined in <a href="#">D2F[5:1]x30</a> .
11:8	<b>IOLimitType.</b> Read-only. 0=16-bit. 1=32-bit.
7:4	<b>IOBase[15:12].</b> Read-write. Lower part of the base address. Upper part is defined in <a href="#">D2F[5:1]x30</a> .
3:0	<b>IOBaseType.</b> Read-only. 0=16-bit. 1=32-bit.

**D2F[5:1]x20 Memory Limit and Base Register**

Reset: 0000\_0000h.

Bits	Description
31:20	<b>MemLimit[31:20]</b> . Read-write.
19:16	<b>MemLimitType</b> . Read-only. 0=32-bit. 1=64-bit.
15:4	<b>MemBase[31:20]</b> . Read-write.
3:0	<b>MemBaseType</b> . Read-only. 0=32-bit. 1=64-bit.

**D2F[5:1]x24 Prefetchable Memory Limit and Base Register**

Reset: 0001\_0001h.

Bits	Description
31:20	<b>PrefMemLimit</b> . Read-write. Lower part of the limit address. Upper part is defined in <a href="#">D2F[5:1]x2C</a> .
19:16	<b>PrefMemLimitType</b> . Read-only. 0=32-bit. 1=64-bit.
15:4	<b>PrefMemBase[31:20]</b> . Read-write. Lower part of the base address. Upper part is defined in <a href="#">D2F[5:1]x28</a> .
3:0	<b>PrefMemBaseType</b> . Read-only. 0=32-bit. 1=64-bit.

**D2F[5:1]x28 Prefetchable Memory Base High Register**

Reset: 0000\_0000h.

Bits	Description
31:0	<b>PrefMemBase[63:32]</b> . Read-write. Upper part of the base address. Lower part is defined in <a href="#">D2F[5:1]x24</a> .

**D2F[5:1]x2C Prefetchable Memory Limit High Register**

Reset: 0000\_0000h.

Bits	Description
31:0	<b>PrefMemLimit[63:32]</b> . Read-write. Upper part of the limit address. Lower part is defined in <a href="#">D2F[5:1]x24</a> .

**D2F[5:1]x30 IO Base and Limit High Register**

Reset: 0000\_0000h.

Bits	Description
31:16	<b>IOLimit[31:16]</b> . Read-write. Upper part of the limit address. Lower part is defined in <a href="#">D2F[5:1]x1C</a> .
15:0	<b>IOBase[31:16]</b> . Read-write. Upper part of the base address. Lower part is defined in <a href="#">D2F[5:1]x1C</a> .

**D2F[5:1]x34 Capabilities Pointer Register**

Reset: 0000\_0050h.

Bits	Description
31:8	Reserved.
7:0	<b>CapPtr: capabilities pointer.</b> Read-only. Pointer to PM capability.

**D2F[5:1]x3C Bridge Control Register**

Reset: 0000\_00FFh.

Bits	Description
31:24	Reserved.
23	<b>FastB2BCap: Fast back-to-back capability.</b> Read-only.
22	<b>SecondaryBusReset: Secondary bus reset.</b> Read-write. Setting this bit triggers a hot reset on the corresponding PCI Express® Port.
21	<b>MasterAbortMode: Master abort mode.</b> Read-only.
20	<b>Vga16En: VGA IO 16 bit decoding enable.</b> 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	<b>VgaEn: VGA enable.</b> 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: Memory accesses in the range of A_0000h to B_FFFFh and 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	<b>IsaEn: ISA enable.</b> Read-write.
17	<b>SerrEn: SERR enable.</b> Read-write.
16	<b>ParityResponseEn: Parity response enable.</b> 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	<b>IntPinR: interrupt pin.</b> Read-only.
10:8	<b>IntPin: interrupt pin.</b> IF (D0F0xE4_x0140_0010[HwInitWrLock]==1) THEN Read-only. ELSE Read-write. ENDIF.
7:0	<b>IntLine: Interrupt line.</b> Read-write.

**D2F[5:1]x50 Power Management Capability Register**

Reset: 0003\_5801h.

Bits	Description
31:27	<b>PmeSupport.</b> Read-only.
26	<b>D2Support: D2 support.</b> Read-only. D2 is not supported
25	<b>D1Support: D1 support.</b> Read-only. D1 is not supported
24:22	<b>AuxCurrent: auxiliary current.</b> Read-only. Auxiliary current is not supported.

21	<b>DevSpecificInit: device specific initialization.</b> Read-only. This field is hardwired to 0 to indicate that there is no device specific initialization necessary.
20	Reserved.
19	<b>PmeClock.</b> Read-only. 0=Indicate that PCI clock is not needed to generate PME messages.
18:16	<b>Version: version.</b> Read-only. 3=PMI Spec 1.2
15:8	<b>NextPtr: next pointer.</b> Read-only. 58h=Address of the next capability structure.
7:0	<b>CapID: capability ID.</b> Read-only. 01h=PCI power management data structure.

### **D2F[5:1]x54 Power Management Control and Status Register**

Bits	Description												
31:24	<b>PmiData.</b> Read-only. Reset: 0.												
23	<b>BusPwrEn.</b> Read-only. Reset: 0.												
22	<b>B2B3Support.</b> Read-only. Reset: 0. B states are not supported.												
21:16	Reserved.												
15	<b>PmeStatus: PME status.</b> 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	<b>DataScale: data scale.</b> Read-only. Reset: 0.												
12:9	<b>DataSelect: data select.</b> Read-only. Reset: 0.												
8	<b>PmeEn: PME# enable.</b> Read-write. Reset: 0.												
7:4	Reserved.												
3	<b>NoSoftReset: no soft reset.</b> Read-only. Reset: 0. Software is required to re-initialize the function when returning from D3 <sub>hot</sub> .												
2	Reserved.												
1:0	<b>PowerState: power state.</b> 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. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Bits</th> <th>Definition</th> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>D0</td> <td>10b</td> <td>Reserved</td> </tr> <tr> <td>01b</td> <td>Reserved</td> <td>11b</td> <td>D3</td> </tr> </tbody> </table>	Bits	Definition	Bits	Definition	00b	D0	10b	Reserved	01b	Reserved	11b	D3
Bits	Definition	Bits	Definition										
00b	D0	10b	Reserved										
01b	Reserved	11b	D3										

### **D2F[5:1]x58 PCI Express® Capability Register**

Reset: 0042\_A010h.

Bits	Description
31:30	Reserved.
29:25	<b>IntMessageNum: interrupt message number.</b> Read-only. This register indicates which MSI vector is used for the interrupt message.
24	<b>SlotImplemented: Slot implemented.</b> Read-only. 1=The IO Link associated with this port is connected to a slot.
23:20	<b>DeviceType: device type.</b> Read-only. 4h=Root complex.

19:16	<b>Version.</b> Read-only. 2h=GEN 2 compliant.
15:8	<b>NextPtr: next pointer.</b> Read-only. A0h=Pointer to the next capability structure.
7:0	<b>CapID: capability ID.</b> Read-only. 10h=PCIe® Capability structure.

### D2F[5:1]x5C Device Capability Register

Reset: 0000\_0020h.

Bits	Description
31:29	Reserved.
28	<b>FlrCapable: function level reset capability.</b> Read-only.
27:26	<b>CapturedSlotPowerScale: captured slot power limit scale.</b> Read-only.
25:18	<b>CapturedSlotPowerLimit: captured slot power limit value.</b> Read-only.
17:16	Reserved.
15	<b>RoleBasedErrReporting: role-based error reporting.</b> Read-only.
14:12	Reserved.
11:9	<b>L1AcceptableLatency: endpoint L1 Acceptable Latency.</b> Read-only.
8:6	<b>L0SAcceptableLatency: endpoint L0s Acceptable Latency.</b> Read-only.
5	<b>ExtendedTag: extended tag support.</b> Read-only. 1: 8 bit tag supported 0: 5 bit tag supported.
4:3	<b>PhantomFunc: phantom function support.</b> Read-only. 0=No phantom functions supported.
2:0	<b>MaxPayloadSupport: maximum supported payload size.</b> Read-only. 000b=128 bytes max payload size.

### D2F[5:1]x60 Device Control and Status Register

Reset: 0000\_2810h.

Bits	Description
31:22	Reserved.
21	<b>TransactionsPending: transactions pending.</b> Read-only. 0=No internally generated non-posted transactions pending.
20	<b>AuxPwr: auxiliary power.</b> Read-only.
19	<b>UsrDetected: unsupported request detected.</b> 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	<b>FatalErr: fatal error detected.</b> 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	<b>NonFatalErr: non-fatal error detected.</b> Read; Write-1-to-clear. 1=The port detected a non-fatal error. Errors are logged in this register even if error reporting is disabled.
16	<b>CorrErr: correctable error detected.</b> 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	<b>BridgeCfgRetryEn: bridge configuration retry enable.</b> Read-only.

14:12	<b>MaxRequestSize: maximum request size.</b> Read-write.																
11	<b>NoSnoopEnable: enable no snoop.</b> 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	<b>AuxPowerPmEn: auxiliary power PM enable.</b> Read-only.																
9	<b>PhantomFuncEn: phantom functions enable.</b> Read-only.																
8	<b>ExtendedTagEn: extended tag enable.</b> Read-write. 1=8-bit tags generation enabled. 0=5-bit tags are used.																
7:5	<b>MaxPayloadSize: maximum supported payload size.</b> Read-write. <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>128B</td> <td>3h</td> <td>1024B</td> </tr> <tr> <td>1h</td> <td>256B</td> <td>4h</td> <td>2048B</td> </tr> <tr> <td>2h</td> <td>512B</td> <td>5h</td> <td>4096B</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	<u>Bits</u>	<u>Definition</u>	0h	128B	3h	1024B	1h	256B	4h	2048B	2h	512B	5h	4096B
<u>Bits</u>	<u>Definition</u>	<u>Bits</u>	<u>Definition</u>														
0h	128B	3h	1024B														
1h	256B	4h	2048B														
2h	512B	5h	4096B														
4	<b>RelaxedOrdEn: relaxed ordering enable.</b> 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	<b>UsrReportEn: unsupported request reporting enable.</b> Read-write. 1=Reporting of unsupported requests enabled.																
2	<b>FatalErrEn: fatal error reporting enable.</b> Read-write. 1=Enable sending ERR_FATAL messages.																
1	<b>NonFatalErrEn: non-fatal error reporting enable.</b> Read-write. 1=Enable sending ERR_NONFATAL messages.																
0	<b>CorrErrEn: correctable error reporting enable.</b> Read-write. 1=Enable sending ERR_CORR messages.																

### D2F[5:1]x64 IO Link Capability Register

Read-only.

Bits	Description
31:24	<b>PortNumber: port number.</b> Reset: 0. This field indicates the port number for the given IO link.
23	Reserved.
22	<b>AspmOptionalityCompliance.</b> Reset: 1. This field indicates if the component supports the ASPM Optionality ECN.
21	<b>LinkBWNotificationCap: link bandwidth notification capability.</b> Reset: 0.
20	<b>DIActiveReportingCapable: data link layer active reporting capability.</b> Reset: 0.
19	<b>SurpriseDownErrReporting.</b> Reset: 0. 1=This field indicates if the component supports the detecting and reporting of a Surprise Down error condition.
18	<b>ClockPowerManagement: clock power management.</b> Reset: 0. 0=Indicates that the reference clock must not be removed while in L1 or L2/L3 ready link states.
17:15	<b>L1ExitLatency: L1 exit latency.</b> Reset: 010b. 010b=Indicate an exit latency between 2 us and 4 us.
14:12	<b>L0sExitLatency: L0s exit latency.</b> Reset: 001b. 001b=Indicates an exit latency between 64 ns and 128 ns.
11:10	<b>PMSupport: active state power management support.</b> Reset: 11b. 11b=Indicates support of L0s and L1.

9:4	<p><b>LinkWidth: maximum link width.</b> Value: 10h.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr><td>00h</td><td>Reserved.</td></tr> <tr><td>01h</td><td>1 lanes</td></tr> <tr><td>02h</td><td>2 lanes</td></tr> <tr><td>03h</td><td>Reserved.</td></tr> <tr><td>04h</td><td>4 lanes</td></tr> <tr><td>07h-05h</td><td>Reserved.</td></tr> <tr><td>08h</td><td>8 lanes</td></tr> <tr><td>0Bh-09h</td><td>Reserved.</td></tr> <tr><td>0Ch</td><td>12 lanes</td></tr> <tr><td>0Fh-0Dh</td><td>Reserved.</td></tr> <tr><td>10h</td><td>16 lanes</td></tr> <tr><td>1Fh-11h</td><td>Reserved.</td></tr> <tr><td>20h</td><td>32 lanes</td></tr> <tr><td>3Fh-21h</td><td>Reserved.</td></tr> </tbody> </table>	Bits	Definition	00h	Reserved.	01h	1 lanes	02h	2 lanes	03h	Reserved.	04h	4 lanes	07h-05h	Reserved.	08h	8 lanes	0Bh-09h	Reserved.	0Ch	12 lanes	0Fh-0Dh	Reserved.	10h	16 lanes	1Fh-11h	Reserved.	20h	32 lanes	3Fh-21h	Reserved.
Bits	Definition																														
00h	Reserved.																														
01h	1 lanes																														
02h	2 lanes																														
03h	Reserved.																														
04h	4 lanes																														
07h-05h	Reserved.																														
08h	8 lanes																														
0Bh-09h	Reserved.																														
0Ch	12 lanes																														
0Fh-0Dh	Reserved.																														
10h	16 lanes																														
1Fh-11h	Reserved.																														
20h	32 lanes																														
3Fh-21h	Reserved.																														
3:0	<p><b>LinkSpeed: link speed.</b> Value:  IF (D2F[5:1]xE4_xA4[LcGen2EnStrap]==0 THEN 1h  ELSEIF (D2F[5:1]xE4_xA4[LcGen2EnStrap]==1 THEN 2h  ENDIF.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr><td>0h</td><td>Reserved.</td></tr> <tr><td>1h</td><td>2.5 Gb/s</td></tr> <tr><td>2h</td><td>5.0 Gb/s</td></tr> <tr><td>Fh-3h</td><td>Reserved.</td></tr> </tbody> </table>	Bits	Definition	0h	Reserved.	1h	2.5 Gb/s	2h	5.0 Gb/s	Fh-3h	Reserved.																				
Bits	Definition																														
0h	Reserved.																														
1h	2.5 Gb/s																														
2h	5.0 Gb/s																														
Fh-3h	Reserved.																														

### D2F[5:1]x68 IO Link Control and Status Register

Reset: 1001\_0000h.

Bits	Description
31	<b>LinkAutonomousBWStatus: link autonomous bandwidth status.</b> IF (D2F[5:1]x64[LinkBWNotificationCap]==0) THEN Read-only. ELSE Read-write; updated-by-hardware. ENDIF.
30	<b>LinkBWManagementStatus: link bandwidth management status.</b> IF (D2F[5:1]x64[LinkBWNotificationCap]==0) THEN Read-only. ELSE Read-write; updated-by-hardware. ENDIF.
29	<b>DLActive: data link layer link active.</b> Read-only; updated-by-hardware. This bit indicates the status of the data link control and management state machine. 1=DL_Active state. 0=All other states.
28	<b>SlotClockCfg: slot clock configuration.</b> Read-only; updated-by-hardware. 1=The root port uses the same clock that the platform provides.
27	<b>LinkTraining: link training.</b> Read-only; updated-by-hardware. 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	<p><b>NegotiatedLinkWidth: negotiated link width.</b> Read-only; updated-by-hardware. This field indicates the negotiated width of the given PCI Express® link.</p> <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>00h</td> <td>Reserved.</td> </tr> <tr> <td>01h</td> <td>1 lanes</td> </tr> <tr> <td>02h</td> <td>2 lanes</td> </tr> <tr> <td>03h</td> <td>Reserved.</td> </tr> <tr> <td>04h</td> <td>4 lanes</td> </tr> <tr> <td>07h-05h</td> <td>Reserved.</td> </tr> <tr> <td>08h</td> <td>8 lanes</td> </tr> <tr> <td>0Bh-09h</td> <td>Reserved.</td> </tr> <tr> <td>0Ch</td> <td>12 lanes</td> </tr> <tr> <td>0Fh-0Dh</td> <td>Reserved.</td> </tr> <tr> <td>10h</td> <td>16 lanes</td> </tr> <tr> <td>1Fh-11h</td> <td>Reserved.</td> </tr> <tr> <td>20h</td> <td>32 lanes</td> </tr> <tr> <td>3Fh-21h</td> <td>Reserved.</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	00h	Reserved.	01h	1 lanes	02h	2 lanes	03h	Reserved.	04h	4 lanes	07h-05h	Reserved.	08h	8 lanes	0Bh-09h	Reserved.	0Ch	12 lanes	0Fh-0Dh	Reserved.	10h	16 lanes	1Fh-11h	Reserved.	20h	32 lanes	3Fh-21h	Reserved.
<u>Bits</u>	<u>Definition</u>																														
00h	Reserved.																														
01h	1 lanes																														
02h	2 lanes																														
03h	Reserved.																														
04h	4 lanes																														
07h-05h	Reserved.																														
08h	8 lanes																														
0Bh-09h	Reserved.																														
0Ch	12 lanes																														
0Fh-0Dh	Reserved.																														
10h	16 lanes																														
1Fh-11h	Reserved.																														
20h	32 lanes																														
3Fh-21h	Reserved.																														
19:16	<p><b>LinkSpeed: link speed.</b> Read-only; updated-by-hardware.</p> <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>00h</td> <td>Reserved.</td> </tr> <tr> <td>01h</td> <td>2.5 Gb/s</td> </tr> <tr> <td>02h</td> <td>5.0 Gb/s</td> </tr> <tr> <td>Fh-3h</td> <td>Reserved.</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	00h	Reserved.	01h	2.5 Gb/s	02h	5.0 Gb/s	Fh-3h	Reserved.																				
<u>Bits</u>	<u>Definition</u>																														
00h	Reserved.																														
01h	2.5 Gb/s																														
02h	5.0 Gb/s																														
Fh-3h	Reserved.																														
15:12	Reserved.																														
11	<p><b>LinkAutonomousBWIntEn: link autonomous bandwidth interrupt enable.</b> Read-write. 1=Enables the generation of an interrupt to indicate that the Link AutonomousBWStatus bit has been set.</p>																														
10	<p><b>LinkBWManagementIntEn: link bandwidth management interrupt enable.</b> Read-write. 1=Enables the generation of an interrupt to indicate that the LinkBWManagementStatus has been set.</p>																														
9	<p><b>HWAutonomousWidthDisable: hardware autonomous width disable.</b> Read-write. 1=Disables hardware from changing the link width for reasons other than attempting to correct unreliable link operation by reducing link width.</p>																														
8	<p><b>ClockPowerManagementEn: clock power management enable.</b> Read-write.</p>																														
7	<p><b>ExtendedSync: extended sync.</b> Read-write. 1=Forces the transmission of additional ordered sets when exiting the L0s state and when in the recovery state.</p>																														
6	<p><b>CommonClockCfg: common clock configuration.</b> 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.</p>																														
5	<p><b>RetrainLink: retrain link.</b> Read-write; cleared-when-done. 1=Initiate link retraining.</p>																														
4	<p><b>LinkDis: link disable.</b> Read-write. 1=Disable link. Writes to this bit are immediately reflected in the value read from the bit, regardless of actual link state.</p>																														
3	<p><b>ReadCplBoundary: read completion boundary.</b> Read-only. 0=64 byte read completion boundary.</p>																														



2	Reserved.												
1:0	<b>PmControl: active state power management enable.</b> Read-write. This field controls the level of ASPM supported on the given IO link.												
	<table> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Disabled.</td> <td>10b</td> <td>L1 Entry Enabled.</td> </tr> <tr> <td>01b</td> <td>L0s Entry Enabled.</td> <td>11b</td> <td>L0s and L1 Entry Enabled.</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	<u>Bits</u>	<u>Definition</u>	00b	Disabled.	10b	L1 Entry Enabled.	01b	L0s Entry Enabled.	11b	L0s and L1 Entry Enabled.
<u>Bits</u>	<u>Definition</u>	<u>Bits</u>	<u>Definition</u>										
00b	Disabled.	10b	L1 Entry Enabled.										
01b	L0s Entry Enabled.	11b	L0s and L1 Entry Enabled.										

**D2F[5:1]x6C Slot Capability Register**

Reset: 0004\_0000h.

Bits	Description												
31:19	<b>PhysicalSlotNumber: physical slot number.</b> Read-write. 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	<b>NoCmdCplSupport: no command completed support.</b> Read-write. 1 =Indicates that this slot does not generate software notification when an issued command is completed by the hot-plug controller.												
17	<b>ElecMechIIPresent: electromechanical interlock present.</b> Read-write. 0=Indicates that a electro-mechanical interlock is not implemented for this slot.												
16:15	<b>SlotPwrLimitScale: slot power limit scale.</b> Read-write. Specifies the scale used for the SlotPwrLimitValue. Range of Values: <table> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>1.0</td> <td>10b</td> <td>0.01</td> </tr> <tr> <td>01b</td> <td>0.1</td> <td>11b</td> <td>0.001</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	<u>Bits</u>	<u>Definition</u>	00b	1.0	10b	0.01	01b	0.1	11b	0.001
<u>Bits</u>	<u>Definition</u>	<u>Bits</u>	<u>Definition</u>										
00b	1.0	10b	0.01										
01b	0.1	11b	0.001										
14:7	<b>SlotPwrLimitValue: slot power limit value.</b> 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	<b>HotplugCapable: hot-plug capability.</b> Read-write. 1=Indicates that this slot is capable of supporting hot-plug operations.												
5	<b>HotplugSurprise: hot-plug surprise.</b> Read-write. 1=Indicates that an adapter present in this slot might be removed from the system without any prior notification.												
4	<b>PwrIndicatorPresent: power indicator present.</b> Read-write. 0=Indicates that a power indicator is not implemented for this slot.												
3	<b>AttnIndicatorPresent: attention indicator present.</b> Read-write. 0=Indicates that a attention indicator is not implemented for this slot.												
2	<b>MrlSensorPresent: manual retention latch sensor present.</b> Read-write. 0=Indicates that a manual retention latch sensor is not implemented for this slot.												
1	<b>PwrControllerPresent: power controller present.</b> Read-write. 0=A power controller is not implemented for this slot.												
0	<b>AttnButtonPresent: attention button present.</b> Read-write. 0=An attention button is not implemented for this slot.												

**D2F[5:1]x70 Slot Control and Status Register**

IF (D2F[5:1]x58[SlotImplemented]==0) THEN Reset: 0040\_0000h. ELSE Reset: 0000\_0000h. ENDIF.

Bits	Description
31:25	Reserved.
24	<b>DIStateChanged: data link layer state change.</b> Read; Write-1-to-clear. This bit is set when the value reported in the D2F[5:1]x68[DIActive] is changed. In response to a data link layer state changed event, software must read D2F[5:1]x68[DIActive] to determine if the link is active before initiating configuration cycles to the hot plugged device.
23	<b>ElecMechIISts: electromechanical interlock status.</b> Read-only.
22	<b>PresenceDetectState: presence detect state.</b> 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. 0=Slot empty. 1=Card present in slot. For root ports not connected to slots (D2F[5:1]x58[SlotImplemented]=0b), this bit returns always 1.
21	<b>MriSensorState.</b> Read-only.
20	<b>CmdCpl: command completed.</b> Read-only.
19	<b>PresenceDetectChanged: presence detect changes.</b> Read; Write-1-to-clear. This bit is set when the value reported in PresenceDetectState is changed.
18	<b>MriSensorChanged.</b> Read; Write-1-to-clear.
17	<b>PwrFaultDetected.</b> Read; Write-1-to-clear.
16	<b>AttnButtonPressed: attention button pressed.</b> Read-only.
15:13	Reserved.
12	<b>DIStateChangedEn: data link layer state changed enable.</b> Read-write. 1=Enables software notification when D2F[5:1]x68[DIActive] is changed.
11	<b>ElecMechIICntl: electromechanical interlock control.</b> Read-only.
10	<b>PwrControllerCntl: power controller control.</b> Read-only.
9:8	<b>PwrIndicatorCntl: power indicator control.</b> Read-only.
7:6	<b>AttnIndicatorControl: attention indicator control.</b> Read-only.
5	<b>HotplugIntrEn: hot-plug interrupt enable.</b> Read-only.
4	<b>CmdCplIntrEn: command complete interrupt enable.</b> Read-only.
3	<b>PresenceDetectChangedEn: presence detect changed enable.</b> Read-only.
2	<b>MriSensorChangedEn: manual retention latch sensor changed enable.</b> Read-only.
1	<b>PwrFaultDetectedEn: power fault detected enable.</b> Read-only.
0	<b>AttnButtonPressedEn: attention button pressed enable.</b> Read-only.

**D2F[5:1]x74 Root Complex Capability and Control Register**

Reset: 0001\_0000h.

Bits	Description
31:17	Reserved.

16	<b>CrSoftVisibility: CRS software visibility.</b> Read-only. 1=Indicates that the root port supports returning configuration request retry status (CRS) completion status to software.
15:5	Reserved.
4	<b>CrSoftVisibilityEn: CRS software visibility enable.</b> Read-write. 1=Enables the root port returning configuration request retry status (CRS) completion status to software.
3	<b>PmIntEn: PME interrupt enable.</b> Read-write. 1=Enables interrupt generation upon receipt of a PME message as reflected $D2F[5:1]x78[PmeStatus]$ . A PME interrupt is also generated if $D2F[5:1]x78[PmeStatus]=1$ and this bit is set by software.
2	<b>SerrOnFatalErrEn: system error on fatal error enable.</b> 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	<b>SerrOnNonFatalErrEn: system error on non-fatal error enable.</b> 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	<b>SerrOnCorrErrEn: system error on correctable error enable.</b> 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.

### D2F[5:1]x78 Root Complex Status Register

Reset: 0000\_0000h.

Bits	Description
31:18	Reserved.
17	<b>PmePending: PME pending.</b> 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	<b>PmeStatus: pme status.</b> 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	<b>PmeRequestorId: pme requestor ID.</b> Read-only. This field indicates the PCI requestor ID of the last PME requestor.

### D2F[5:1]x7C Device Capability 2

Reset: 0000\_0000h.

Bits	Description												
31:24	Reserved.												
23:22	<b>MaxEndEndTlpPrefixes: Max number of End-End TLP prefixes supported.</b> Read-only. IF $(D2F[5:1]x7C[EndEndTlpPrefixSupported]==0)$ THEN Reserved. ENDIF. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Bits</th> <th>Definition</th> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>4 End-End TLP Prefixes.</td> <td>10b</td> <td>2 End-End TLP Prefixes.</td> </tr> <tr> <td>01b</td> <td>1 End-End TLP Prefix.</td> <td>11b</td> <td>3 End-End TLP Prefixes.</td> </tr> </tbody> </table>	Bits	Definition	Bits	Definition	00b	4 End-End TLP Prefixes.	10b	2 End-End TLP Prefixes.	01b	1 End-End TLP Prefix.	11b	3 End-End TLP Prefixes.
Bits	Definition	Bits	Definition										
00b	4 End-End TLP Prefixes.	10b	2 End-End TLP Prefixes.										
01b	1 End-End TLP Prefix.	11b	3 End-End TLP Prefixes.										
21	<b>EndEndTlpPrefixSupported: End-End TLP Prefix supported.</b> Read-only.												

20	<b>ExtendedFmtFieldSupported.</b> Read-only. 1=Supports the 3-bit definition of the Fmt field. 0=Supports the 2-bit definition of the Fmt field. Must be set for functions that support End-End TLP prefixes.
19:18	<b>ObffSupported: optimized buffer flush/fill supported.</b> Read-only.
17:14	Reserved.
13:12	Reserved.
11	<b>LtrSupported: latency tolerance supported.</b> Read-only.
10	Reserved.
9:6	Reserved.
5	<b>AriForwardingSupported.</b> Read-only.
4	<b>CplTimeoutDisSup: completion timeout disable supported.</b> Read-only.
3:0	<b>CplTimeoutRangeSup: completion timeout range supported.</b> Read-only. Fh=Completion timeout range is 64s to 50us.

**D2F[5:1]x80 Device Control and Status 2**

Reset: 0000\_8000h.

Bits	Description																																
31:16	Reserved.																																
15	<b>EndEndTlpPrefixBlocking.</b> Read-only. 1=Forwarding of End-End TLP Prefixes is not supported. This bit is hardwired to 1b.																																
14:13	<b>ObffEn: optimized buffer flush/fill enable.</b> Read-write.																																
12:11	Reserved.																																
10	<b>LtrEn: latency tolerance reporting enable.</b> Read-write.																																
9	Reserved.																																
8	Reserved.																																
7:6	Reserved.																																
5	<b>AriForwardingEn.</b> Read-write.																																
4	<b>CplTimeoutDis: completion timeout disable.</b> Read-write. 1=Completion timeout disabled.																																
3:0	<p><b>CplTimeoutValue: completion timeout value.</b> Read-write. BIOS: 6h.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Timeout Range</th> <th>Bits</th> <th>Timeout Range</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>50ms-50us</td> <td>9h</td> <td>900ms-260ms</td> </tr> <tr> <td>1h</td> <td>100us-50us</td> <td>Ah</td> <td>3.5s-1s</td> </tr> <tr> <td>2h</td> <td>10ms-1ms</td> <td>Ch-Bh</td> <td>Reserved</td> </tr> <tr> <td>4h-3h</td> <td>Reserved</td> <td>Dh</td> <td>13s-4s</td> </tr> <tr> <td>5h</td> <td>55ms-16ms</td> <td>Eh</td> <td>64s-4s</td> </tr> <tr> <td>6h</td> <td>210ms-65ms</td> <td>Fh</td> <td>Reserved</td> </tr> <tr> <td>8h-7h</td> <td>Reserved</td> <td></td> <td></td> </tr> </tbody> </table>	Bits	Timeout Range	Bits	Timeout Range	0h	50ms-50us	9h	900ms-260ms	1h	100us-50us	Ah	3.5s-1s	2h	10ms-1ms	Ch-Bh	Reserved	4h-3h	Reserved	Dh	13s-4s	5h	55ms-16ms	Eh	64s-4s	6h	210ms-65ms	Fh	Reserved	8h-7h	Reserved		
Bits	Timeout Range	Bits	Timeout Range																														
0h	50ms-50us	9h	900ms-260ms																														
1h	100us-50us	Ah	3.5s-1s																														
2h	10ms-1ms	Ch-Bh	Reserved																														
4h-3h	Reserved	Dh	13s-4s																														
5h	55ms-16ms	Eh	64s-4s																														
6h	210ms-65ms	Fh	Reserved																														
8h-7h	Reserved																																

**D2F[5:1]x84 IO Link Capability 2**

Bits	Description
31:9	Reserved.
8	<b>CrossLinkSupported</b> . Read-only. Reset: 0.
7:3	Reserved.
2:1	<b>SupportedLinkSpeed</b> . Read-only. Reset: 3h. Specifies the supported link speeds. Bit 1=2.5GT/s, Bit 2=5.0GT/s.
0	Reserved.

**D2F[5:1]x88 IO Link Control and Status 2**

Bits	Description																								
31:22	Reserved.																								
21	<b>LinkEqualizationRequest</b> . Read; write-1-to-clear. Reset: 0. 1=Hardware requests link equalization to be performed.																								
20	<b>EqualizationPhase3Success</b> . Read-only. Reset: 0. 1=Phase 3 of the Transmitter Equalization procedure has completed successfully. Write 1 to clear.																								
19	<b>EqualizationPhase2Success</b> . Read-only. Reset: 0. 1=Phase 2 of the Transmitter Equalization procedure has completed successfully. Write 1 to clear.																								
18	<b>EqualizationPhase1Success</b> . Read-only. Reset: 0. 1=Phase 1 of the Transmitter Equalization procedure has completed successfully. Write 1 to clear.																								
17	<b>EqualizationComplete</b> . Read-only. Reset: 0. 1=Transmitter Equalization procedure has completed. Write 1 to clear.																								
16	<b>CurDeemphasisLevel: current deemphasis level</b> . Read-only. Reset: <a href="#">D2F[5:1]xE4_xA4[LcGen2EnStrap]</a> . 1=-3.5 dB. 0=-6 dB																								
15:12	<b>ComplianceDeemphasis: compliance deemphasis</b> . Read-write. Reset: 0. In Gen2 this field defines the compliance deemphasis level when EnterCompliance is set. Software should leave this field in its default state. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>DeEmph=-6 dB, Preshoot=0 dB</td> </tr> <tr> <td>1h</td> <td>DeEmph=-3.5 dB, Preshoot=0 dB</td> </tr> <tr> <td>2h</td> <td>DeEmph=-4.5 dB, Preshoot=0 dB</td> </tr> <tr> <td>3h</td> <td>DeEmph=-2.5 dB, Preshoot=0 dB</td> </tr> <tr> <td>4h</td> <td>DeEmph=-0 dB, Preshoot=0 dB</td> </tr> <tr> <td>5h</td> <td>DeEmph=-0 dB, Preshoot=2 dB</td> </tr> <tr> <td>6h</td> <td>DeEmph=-0 dB, Preshoot=2.5 dB</td> </tr> <tr> <td>7h</td> <td>DeEmph=-6 dB, Preshoot=3.5 dB</td> </tr> <tr> <td>8h</td> <td>DeEmph=-3.5 dB, Preshoot=3.5 dB</td> </tr> <tr> <td>9h</td> <td>DeEmph=-0 dB, Preshoot=3.5 dB</td> </tr> <tr> <td>Fh-Ah</td> <td>Reserved.</td> </tr> </tbody> </table>	Bits	Definition	0h	DeEmph=-6 dB, Preshoot=0 dB	1h	DeEmph=-3.5 dB, Preshoot=0 dB	2h	DeEmph=-4.5 dB, Preshoot=0 dB	3h	DeEmph=-2.5 dB, Preshoot=0 dB	4h	DeEmph=-0 dB, Preshoot=0 dB	5h	DeEmph=-0 dB, Preshoot=2 dB	6h	DeEmph=-0 dB, Preshoot=2.5 dB	7h	DeEmph=-6 dB, Preshoot=3.5 dB	8h	DeEmph=-3.5 dB, Preshoot=3.5 dB	9h	DeEmph=-0 dB, Preshoot=3.5 dB	Fh-Ah	Reserved.
Bits	Definition																								
0h	DeEmph=-6 dB, Preshoot=0 dB																								
1h	DeEmph=-3.5 dB, Preshoot=0 dB																								
2h	DeEmph=-4.5 dB, Preshoot=0 dB																								
3h	DeEmph=-2.5 dB, Preshoot=0 dB																								
4h	DeEmph=-0 dB, Preshoot=0 dB																								
5h	DeEmph=-0 dB, Preshoot=2 dB																								
6h	DeEmph=-0 dB, Preshoot=2.5 dB																								
7h	DeEmph=-6 dB, Preshoot=3.5 dB																								
8h	DeEmph=-3.5 dB, Preshoot=3.5 dB																								
9h	DeEmph=-0 dB, Preshoot=3.5 dB																								
Fh-Ah	Reserved.																								
11	<b>ComplianceSOS: compliance SOS</b> . Read-write. Reset: 0. 1=The device transmits skip ordered sets in between the modified compliance pattern.																								

10	<b>EnterModCompliance: enter modified compliance.</b> Read-write. Reset: 0. 1=The device transmits modified compliance pattern. Software should leave this field in its default state.										
9:7	<b>XmitMargin: transmit margin.</b> Read-write. Reset: 0. This field controls the non-deemphasized voltage level at the transmitter pins. Software should leave this field in its default state.										
6	<b>SelectableDeemphasis: selectable deemphasis.</b> Read-only. Reset: <a href="#">D2F[5:1]xE4_xA4[LcGen2EnStrap]</a> . 0=Selectable deemphasis is not supported. 1=Selectable deemphasis supported.										
5	<b>HwAutonomousSpeedDisable: hardware autonomous speed disable.</b> Read-write. Cold reset: 0. 1=Support for hardware changing the link speed for device specific reasons disabled.										
4	<b>EnterCompliance: enter compliance.</b> Read-write. Cold reset: 0. 1=Force the link to enter the compliance mode.										
3:0	<p><b>TargetLinkSpeed: target link speed.</b> Read-write. Reset: 2h.                      This field 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.</p> <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Reserved</td> </tr> <tr> <td>1h</td> <td>2.5GT/s</td> </tr> <tr> <td>2h</td> <td>5.0GT/s</td> </tr> <tr> <td>Fh-3h</td> <td>Reserved</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	0h	Reserved	1h	2.5GT/s	2h	5.0GT/s	Fh-3h	Reserved
<u>Bits</u>	<u>Definition</u>										
0h	Reserved										
1h	2.5GT/s										
2h	5.0GT/s										
Fh-3h	Reserved										

**D2F[5:1]x8C Slot Capability 2**

Reset: 0000\_0000h.

Bits	Description
31:0	Reserved.

**D2F[5:1]x90 Slot Control and Status 2**

Reset: 0000\_0000h.

Bits	Description
31:0	Reserved.

**D2F[5:1]xA0 MSI Capability Register**

Reset: 0000\_B005h.

Bits	Description
31:24	Reserved.
23	<b>Msi64bit: MSI 64 bit capability.</b> Read-only. 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	<b>MsiMultiEn: MSI multiple message enable.</b> Read-write. 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	<b>MsiMultiCap: MSI multiple message capability.</b> Read-only. 000b=The device is requesting one vector.
16	<b>MsiEn: MSI enable.</b> Read-write. 1=MSI generation is enabled and INTx generation is disabled. 0=MSI generation disabled and INTx generation is enabled.
15:8	<b>NextPtr: next pointer.</b> Read-only.
7:0	<b>CapID: capability ID.</b> Read-only. 05h=MSI capability structure.

**D2F[5:1]xA4 MSI Message Address Low**

Reset: 0000\_0000h.

Bits	Description
31:2	<b>MsiMsgAddrLo: MSI message address.</b> Read-write. This register specifies the dword aligned address for the MSI memory write transaction.
1:0	Reserved.

**D2F[5:1]xA8 MSI Message Address High**

Reset: 0000\_0000h.

Bits	Description
31:0	<b>MsiMsgAddrHi: MSI message address.</b> Read-write. This register specifies the upper 32-bits of the MSI address.

**D2F[5:1]xAC MSI Message Data**

Reset: 0000\_0000h.

Bits	Description
31:16	Reserved.
15:0	<b>MsiData: MSI message data.</b> Read-write. This register specifies lower 16 bits of data for the MSI memory write transaction. The upper 16 bits are always 0.

**D2F[5:1]xB0 Subsystem and Subvendor Capability ID Register**

Reset: 0000\_B80Dh.

Bits	Description
31:16	Reserved.
15:8	<b>NextPtr: next pointer.</b> Read-only.
7:0	<b>CapID: capability ID.</b> Read-only.

**D2F[5:1]xB4 Subsystem and Subvendor ID Register**

Reset: 0000\_0000h.

Bits	Description
31:16	<b>SubsystemID</b> . Read-only.
15:0	<b>SubsystemVendorID</b> . Read-only.

**D2F[5:1]xB8 MSI Capability Mapping**

Reset: A803\_0008h.

Bits	Description
31:27	<b>CapType: capability type</b> . Read-only.
26:18	Reserved.
17	<b>FixD</b> . Read-only.
16	<b>En</b> . Read-only.
15:8	<b>NextPtr: next pointer</b> . Read-only.
7:0	<b>CapID: capability ID</b> . Read-only.

**D2F[5:1]xBC MSI Mapping Address Low**

Bits	Description
31:20	<b>MsiMapAddrLo</b> . Read-only. Reset: 0. Lower 32-bits of the MSI address.
19:0	Reserved.

**D2F[5:1]xC0 MSI Mapping Address High**

Bits	Description
31:0	<b>MsiMapAddrHi</b> . Read-only. Reset: 0. Upper 32-bits of the MSI address.

**D2F[5:1]xE0 Root Port Index**

Reset: 0000\_0000h.

The index/data pair registers, [D2F\[5:1\]xE0](#) and [D2F\[5:1\]xE4](#), are used to access the registers at [D2F\[5:1\]xE4\\_x\[FF:00\]](#). To access any of these registers, the address is first written into the index register, [D2F\[5:1\]xE0](#), and then the data is read from or written to the data register, [D2F\[5:1\]xE4](#).

Bits	Description
31:8	Reserved.
7:0	<b>PcieIndex</b> . Read-write.



**D2F[5:1]xE4 Root Port Data**

See D2F[5:1]xE0. Address: D2F[5:1]xE0[PCieIndex].

Bits	Description
31:0	<b>PcieData.</b> Read-write.

**D2F[5:1]xE4\_x20 Root Port TX Control**

Reset: 0050\_8000h.

Bits	Description
31:16	Reserved.
15	<b>TxFlushTlpDis: TLP flush disable.</b> Read-write. 1=Disable flushing TLPs when the link is down.
14:0	Reserved.

**D2F[5:1]xE4\_x50 Root Port Lane Status**

Reset: 0000\_0000h.

Bits	Description																				
31:7	Reserved.																				
6:1	<b>PhyLinkWidth: port link width.</b> Read-only; updated-by-hardware. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00_0000b</td> <td>disabled</td> <td>00_1000b</td> <td>x8</td> </tr> <tr> <td>00_0001b</td> <td>x1</td> <td>01_0000b</td> <td>x12</td> </tr> <tr> <td>00_0010b</td> <td>x2</td> <td>10_0000b</td> <td>x16</td> </tr> <tr> <td>00_0100b</td> <td>x4</td> <td></td> <td></td> </tr> </tbody> </table>	Bits	Definition	Bits	Definition	00_0000b	disabled	00_1000b	x8	00_0001b	x1	01_0000b	x12	00_0010b	x2	10_0000b	x16	00_0100b	x4		
Bits	Definition	Bits	Definition																		
00_0000b	disabled	00_1000b	x8																		
00_0001b	x1	01_0000b	x12																		
00_0010b	x2	10_0000b	x16																		
00_0100b	x4																				
0	<b>PortLaneReversal: port lane reversal.</b> Read-only. 1=Port lanes order is reversed.																				

**D2F[5:1]xE4\_x6A Root Port Error Control**

Reset: 0000\_0500h.

Bits	Description
31:1	Reserved.
0	<b>ErrReportingDis: advanced error reporting disable.</b> Read-write. BIOS: 1. 1=Error reporting disabled. 0=Error reporting enabled.

**D2F[5:1]xE4\_x70 Root Port Receiver Control**

Reset: 0188\_4000h.

Bits	Description
31:20	Reserved.
19	<b>RxRcbCplTimeoutMode: RCB completion timeout mode.</b> Read-write. BIOS: 1. 1=Timeout on link down.

18:16	<b>RxRcbCplTimeout: RCB completion timeout.</b> Read-write.			
	<u>Bits</u>	<u>Definition</u>	<u>Bits</u>	<u>Definition</u>
	000b	Disabled	100b	50ms
	001b	50us	101b	100ms
	010b	10ms	110b	500ms
	011b	25ms	111b	1ms
15:0	Reserved.			

**D2F[5:1]xE4\_xA0 Per Port Link Controller (LC) Control**

Reset: 4000\_0030h.

Bits	Description			
31:24	Reserved.			
23	<b>LcL1ImmediateAck: immediate ACK ASPM L1 entry.</b> Read-write. BIOS: 1. 1=Always ACK ASPM L1 entry DLLPs.			
22:16	Reserved.			
15:12	<b>LcL1Inactivity: L1 inactivity timer.</b> Read-write.			
	<u>Bits</u>	<u>Definition</u>	<u>Bits</u>	<u>Definition</u>
	0h	L1 disabled	8h	400us
	1h	1us	9h	1ms
	2h	2us	Ah	40us
	3h	4us	Bh	10ms
	4h	10us	Ch	40ms
	5h	20us	Dh	100ms
	6h	40us	Eh	400ms
	7h	100us	Fh	reserved
11:8	<b>LcL0sInactivity: L0s inactivity timer.</b> Read-write.			
	<u>Bits</u>	<u>Definition</u>	<u>Bits</u>	<u>Definition</u>
	0h	L0s disabled	8h	4us
	1h	40ns	9h	10us
	2h	80ns	Ah	40us
	3h	120ns	Bh	100us
	4h	200ns	Ch	400us
	5h	400ns	Dh	1ms
	6h	1us	Eh	4ms
	7h	2us	Fh	reserved
7:4	<b>Lc16xClearTxPipe.</b> Read-write. BIOS: <u>1h</u> . Specifies the number of clocks to drain the TX pipe.			
3:0	Reserved.			

**D2F[5:1]xE4\_xA1 LC Training Control**

Reset: 5400\_1880h.

Bits	Description
31:12	Reserved.

11	<b>LcDontGotoL0sifL1Armed: prevent Ls0 entry is L1 request in progress.</b> 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.

**D2F[5:1]xE4\_xA2 LC Link Width Control**

Reset: 00A0\_0006h.

Bits	Description																				
31:24	Reserved.																				
23	Reserved.																				
22:21	<b>LcDynLanesPwrState: unused link power state.</b> Read-write. Controls the state of unused links after a reconfiguration. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Bits</th> <th>Definition</th> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>on</td> <td>10b</td> <td>SB2</td> </tr> <tr> <td>01b</td> <td>SB1</td> <td>11b</td> <td>Off</td> </tr> </tbody> </table>	Bits	Definition	Bits	Definition	00b	on	10b	SB2	01b	SB1	11b	Off								
Bits	Definition	Bits	Definition																		
00b	on	10b	SB2																		
01b	SB1	11b	Off																		
20	<b>LcUpconfigCapable: upconfigure capable.</b> Read-only; updated-by-hardware. 1=Both ends of the link are upconfigure capable. 0=Both ends of the link are not upconfigure capable.																				
19:14	Reserved.																				
13	<b>LcUpconfigureDis: upconfigure disable.</b> Read-write. 1=Disable link upconfigure.																				
12	<b>LcUpconfigureSupport: upconfigure support.</b> Read-write.																				
11	<b>LcShortReconfigEn: short re-configuration enable.</b> Read-write. 1=Enable short link re-configuration																				
10	<b>LcRenegotiateEn: link reconfiguration enable.</b> Read-write. 1=Enable link re-negotiation.																				
9	<b>LcRenegotiationSupport: re-negotiation support.</b> Read-only; updated-by-hardware. 1=Link re-negotiation not supported by the downstream device.																				
8	<b>LcReconfigNow: re-configure link.</b> Read-write; cleared-when-done. 1=Initiate link width change.																				
7	<b>LcReconfigArcMissingEscape.</b> Read-write. 1=Expedite transition from Recovery.Idle to Detect during a long reconfiguration.																				
6:4	<b>LcLinkWidthRd: current link width.</b> Read-only; updated-by-hardware. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Bits</th> <th>Definition</th> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>0</td> <td>100b</td> <td>8</td> </tr> <tr> <td>001b</td> <td>1</td> <td>101b</td> <td>12</td> </tr> <tr> <td>010b</td> <td>2</td> <td>110b</td> <td>16</td> </tr> <tr> <td>011b</td> <td>4</td> <td>111b</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Definition	Bits	Definition	000b	0	100b	8	001b	1	101b	12	010b	2	110b	16	011b	4	111b	Reserved
Bits	Definition	Bits	Definition																		
000b	0	100b	8																		
001b	1	101b	12																		
010b	2	110b	16																		
011b	4	111b	Reserved																		
3	Reserved.																				
2:0	<b>LcLinkWidth: link width required.</b> Read-write. See: LcLinkWidthRd.																				

**D2F[5:1]xE4\_xA3 LC Number of FTS Control**

Reset: 00FF\_020Ch.

Bits	Description
31:10	Reserved.

9	<b>LcXmitFtsBeforeRecovery:</b> transmit FTS before recovery. Read-write. 1=Transmit FTS before recovery.
8:0	Reserved.

**D2F[5:1]xE4\_xA4 LC Link Speed Control**

Reset: 0440\_0100h.

Bits	Description
31:28	Reserved.
27	<b>LcMultUpstreamAutoSpdChngEn:</b> enable multiple automatic speed changes. Read-write. 1=Enable multiple automatic speed changes when D2F[5:1]xE4_xC0[StrapAutoRcSpeedNegotiationDis]=0 and no failures occurred in previous speed change attempts.
26:20	Reserved.
19	<b>LcOtherSideSupportsGen2:</b> downstream link supports gen2. Read-only; updated-by-hardware. 1=The downstream link currently supports gen2.
18:15	Reserved.
14:13	Reserved.
12	<b>LcSpeedChangeAttemptFailed:</b> speed change attempt failed. Read-only; updated-by-hardware. 1=LcSpeedChangeAttemptsAllowed has been reached.
11:10	Reserved.
9	<b>LcInitiateLinkSpeedChange:</b> initiate link speed change. Read-write; cleared-when-done. 1=Initiate link speed negotiation.
8:7	Reserved.
6	<b>LcForceDisSwSpeedChange:</b> force disable software speed changes. Read-write. 1=Force the PCIe <sup>®</sup> core to disable speed changes initiated by private registers.
5:1	Reserved.
0	<b>LcGen2EnStrap:</b> Gen2 PCIe <sup>®</sup> support enable. Read-write. 1=Gen2 PCIe <sup>®</sup> support enabled. 0=Gen2 PCIe <sup>®</sup> support disabled.

**D2F[5:1]xE4\_xA5 LC State 0**

Cold reset: 0000\_0000h.

Bits	Description
31:30	Reserved.
29:24	<b>LcPrevState3:</b> previous link state 3. Read-only; updated-by-hardware. See: <a href="#">Table 100 [Link controller state encodings]</a> .
23:22	Reserved.
21:16	<b>LcPrevState2:</b> previous link state 2. Read-only; updated-by-hardware. See: <a href="#">Table 100 [Link controller state encodings]</a>
15:14	Reserved.
13:8	<b>LcPrevState1:</b> previous link state 1. Read-only; updated-by-hardware. See: <a href="#">Table 100 [Link controller state encodings]</a> .

7:6	Reserved.
5:0	<b>LcCurrentState: current link state.</b> Read-only; updated-by-hardware. See: <a href="#">Table 100 [Link controller state encodings]</a> .

**Table 100: 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.		

**D2F[5:1]xE4\_xB1 LC Control 2**

Reset: 8608\_0280h.

Bits	Description
31:21	Reserved.
20	<b>LcBlockEIdleinL0: block electrical idle in l0.</b> Read-write. BIOS: 1. 1=Prevent electrical idle from causing the receiver to transition from L0 to L0s.
19	<b>LcDeassertRxEnInL0s: deassert RX_EN in L0s.</b> Read-write. 1=Turn off transmitters in L0s.
18:0	Reserved.

**D2F[5:1]xE4\_xB5 LC Control 3**

Reset: 2850\_5020h.

Bits	Description
31	Reserved.

30	<b>LcGoToRecovery: go to recovery.</b> Read-write. 1=Force link in the L0 state to transition to the Recovery state.										
29:4	Reserved.										
3	<b>LcRcvdDeemphasis: received deemphasis.</b> Read-only; updated-by-hardware. Deemphasis advertised by the downstream device. 1=3.5dB. 0=6dB.										
2:1	<b>LcSelectDeemphasisCntl: deemphasis control.</b> Read-write. Specifies the deemphasis used by the transmitter. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Use deemphasis from LcSelectDeemphasis.</td> </tr> <tr> <td>01b</td> <td>Use deemphasis advertised by the downstream device.</td> </tr> <tr> <td>10b</td> <td>6dB</td> </tr> <tr> <td>11b</td> <td>3.5dB</td> </tr> </tbody> </table>	Bits	Definition	00b	Use deemphasis from LcSelectDeemphasis.	01b	Use deemphasis advertised by the downstream device.	10b	6dB	11b	3.5dB
Bits	Definition										
00b	Use deemphasis from LcSelectDeemphasis.										
01b	Use deemphasis advertised by the downstream device.										
10b	6dB										
11b	3.5dB										
0	<b>LcSelectDeemphasis: downstream deemphasis.</b> Read-write. Specifies the downstream deemphasis. 1=3.5dB. 0=6dB.										

**D2F[5:1]xE4\_xC0 LC Strap Override**

Reset: 0000\_0000h.

Bits	Description
31:16	Reserved.
15	<b>StrapAutoRcSpeedNegotiationDis: autonomous speed negotiation disable strap override.</b> Read-write. 1=Disable autonomous root complex speed negotiation to Gen2.
14	Reserved.
13	<b>StrapForceCompliance: force compliance strap override.</b> Read-write.
12:0	Reserved.

**D2F[5:1]xE4\_xC1 Root Port Miscellaneous Strap Override**

Reset: 0000\_0000h.

Bits	Description
31:6	Reserved.
5	<b>StrapLtrSupported.</b> Read-write.
4:3	<b>StrapObffSupported.</b> Read-write.
2	<b>StrapExtendedFmtSupported: Extended Fmt Supported strap override.</b> Read-write.
1	<b>StrapE2EPrefixEn: E2E Prefix En strap override.</b> Read-write.
0	<b>StrapReverseLanes: reverse lanes strap override.</b> Read-write.

**D2F[5:1]xE4\_xD0 Root Port ECC Skip OS Feature**

Reset: 0000\_0100h.

Bits	Description
31:16	<b>BchEccErrorStatus.</b> Read-write. Indicates that lane errors are above the specified threshold. (One bit per lane.)
15:8	<b>BchEccErrorThreshold.</b> Read-write. Error threshold.
7:1	Reserved.
0	<b>StrapBchEccEn.</b> Read-write.

**D2F[5:1]x100 Vendor Specific Enhanced Capability Register**

Bits	Description
31:20	<b>NextPtr: next pointer.</b> Read-only. IF (D0F0xE4_x0140_00B0[StrapF0AerEn] == 1) THEN Reset: 150h. ELSE Reset: 000h. ENDIF.
19:16	<b>CapVer: capability version.</b> Read-only. Reset: 1h.
15:0	<b>CapID: capability ID.</b> Read-only. Reset: Bh.

**D2F[5:1]x104 Vendor Specific Header Register**

Reset: 0101\_0001h.

Bits	Description
31:20	<b>VsecLen: vendor specific enhanced capability structure length.</b> Read-only. Defined the number of bytes of the entire vendor specific enhanced capability structure including the header.
19:16	<b>VsecRev: vendor specific enhanced capability version.</b> Read-only.
15:0	<b>VsecID: vendor specific enhanced capability ID.</b> Read-only.

**D2F[5:1]x108 Vendor Specific 1 Register**

Reset: 0000\_0000h.

Bits	Description
31:0	<b>Scratch: scratch.</b> Read-write. This field does not control any hardware.

**D2F[5:1]x10C Vendor Specific 2 Register**

Reset: 0000\_0000h.

Bits	Description
31:0	<b>Scratch: scratch.</b> Read-write. This field does not control any hardware.

**D2F[5:1]x128 Virtual Channel 0 Resource Status Register**

Reset: 0002\_0000h.

Bits	Description
31:18	Reserved.
17	<b>VcNegotiationPending: virtual channel negotiation pending.</b> Read-only; updated-by-hardware. 1=Virtual channel negotiation in progress. This bit must be 0 before the virtual channel can be used.
16	<b>PortArbTableStatus: port arbitration table status.</b> Read-only.
15:0	Reserved.

**D2F[5:1]x150 Advanced Error Reporting Capability**

Bits	Description
31:20	<b>NextPtr: next pointer.</b> Read-only. IF (D0F0xE4_x0140_00B0[StrapF0AcsEn] == 1) THEN 2A0h. ELSE Reset: 000h. ENDIF.
19:16	<b>CapVer: capability version.</b> Read-only. Reset: 2h.
15:0	<b>CapID: capability ID.</b> Read-only. Reset: 1h.

**D2F[5:1]x154 Uncorrectable Error Status**

Cold reset: 0000\_0000h.

Bits	Description
31:26	Reserved.
25	<b>TlpPrefixStatus: TLP prefix blocked status.</b> Read; Write-1-to-clear.
24	<b>AtomicOpEgressBlockedTLPStatus: atomic op egress blocked TLP status.</b> Read; Write-1-to-clear.
23	<b>McBlockedTLPStatus: MC blocked TLP status.</b> Read; Write-1-to-clear.
22	<b>UncorrInternalErrStatus: uncorrectable internal error status.</b> Read; Write-1-to-clear.
21	<b>AcsViolationStatus: access control service status.</b> Read; Write-1-to-clear.
20	<b>UnsuppReqErrStatus: unsupported request error status.</b> Read; Write-1-to-clear. The header of the unsupported request is logged.
19	<b>EcrcErrStatus: end-to-end CRC error status.</b> Read; Write-1-to-clear.
18	<b>MalTlpStatus: malformed TLP status.</b> Read; Write-1-to-clear. The header of the malformed TLP is logged.
17	<b>RcvOvflStatus: receiver overflow status.</b> Read-only.
16	<b>UnexpCplStatus: unexpected completion timeout status.</b> Read; Write-1-to-clear. The header of the unexpected completion is logged.
15	<b>CplAbortErrStatus: completer abort error status.</b> Read; Write-1-to-clear.
14	<b>CplTimeoutStatus: completion timeout status.</b> Read; Write-1-to-clear.



13	<b>FcErrStatus: flow control error status.</b> Read-only.
12	<b>PsnErrStatus: poisoned TLP status.</b> Read; Write-1-to-clear. The header of the poisoned transaction layer packet is logged.
11:6	Reserved.
5	<b>SurprdnErrStatus: surprise down error status.</b> Read-only. 0=Detection and reporting of surprise down errors is not supported.
4	<b>DlpErrStatus: data link protocol error status.</b> Read; Write-1-to-clear.
3:0	Reserved.

### D2F[5:1]x158 Uncorrectable Error Mask

Cold reset: 0000\_0000h.

Bits	Description
31:26	Reserved.
25	<b>TlpPrefixMask: TLP prefix blocked mask.</b> Read-only.
24	<b>AtomicOpEgressBlockedTLPMask: atomic op egress blocked TLP mask.</b> Read-only.
23	<b>McBlockedTLPMask: MC blocked TLP mask.</b> Read-only.
22	<b>UncorrInternalErrMask: uncorrectable internal error mask.</b> Read-write.
21	<b>AcsViolationMask: access control service mask.</b> Read-only. 1=ACS violation errors are not reported.
20	<b>UnsuppReqErrMask: unsupported request error mask.</b> Read-write. 1=Unsupported request errors are not reported.
19	<b>EcrcErrMask: end-to-end CRC error mask.</b> Read-write.
18	<b>MalTlpMask: malformed TLP mask.</b> Read-write. 1=Malformed TLP errors are not reported.
17	<b>RcvOvflMask: receiver overflow mask.</b> Read-only.
16	<b>UnexpCplMask: unexpected completion timeout mask.</b> Read-write. 1=Unexpected completion errors are not reported.
15	<b>CplAbortErrMask: completer abort error mask.</b> Read-write.
14	<b>CplTimeoutMask: completion timeout mask.</b> Read-write. 1=Completion timeout errors are not reported.
13	<b>FcErrMask: flow control error mask.</b> Read-only.
12	<b>PsnErrMask: poisoned TLP mask.</b> Read-write. 1=Poisoned TLP errors are not reported.
11:6	Reserved.
5	<b>SurprdnErrMask: surprise down error mask.</b> Read-only.
4	<b>DlpErrMask: data link protocol error mask.</b> Read-write. 1=Data link protocol errors are not reported.
3:0	Reserved.

**D2F[5:1]x15C Uncorrectable Error Severity**

Cold reset: 0006\_2030h.

Bits	Description
31:26	Reserved.
25	<b>TlpPrefixSeverity: TLP prefix blocked severity.</b> Read-only.
24	<b>AtomicOpEgressBlockedTLPSeverity: atomic op egress blocked TLP severity.</b> Read-only.
23	<b>McBlockedTLPSeverity: MC blocked TLP severity.</b> Read-only.
22	<b>UncorrInternalErrSeverity: uncorrectable internal error severity.</b> Read-only.
21	<b>AcsViolationSeverity: access control service severity.</b> Read-only. 1=Fatal error. 0=Non-fatal error.
20	<b>UnsuppReqErrSeverity: unsupported request error severity.</b> Read-write. 1=Fatal error. 0=Non-fatal error.
19	<b>EcreErrSeverity: end-to-end CRC error severity.</b> Read-only.
18	<b>MalTlpSeverity: malformed TLP severity.</b> Read-write. 1=Fatal error. 0=Non-fatal error.
17	<b>RcvOvflSeverity: receiver overflow severity.</b> Read-only.
16	<b>UnexpCplSeverity: unexpected completion timeout severity.</b> Read-write. 1=Fatal error. 0=Non-fatal error.
15	<b>CplAbortErrSeverity: completer abort error severity.</b> Read-only.
14	<b>CplTimeoutSeverity: completion timeout severity.</b> Read-write. 1=Fatal error. 0=Non-fatal error.
13	<b>FcErrSeverity: flow control error severity.</b> Read-only.
12	<b>PsnErrSeverity: poisoned TLP severity.</b> Read-write. 1=Fatal error. 0=Non-fatal error.
11:6	Reserved.
5	<b>SurprdnErrSeverity: surprise down error severity.</b> Read-only.
4	<b>DlpErrSeverity: data link protocol error severity.</b> Read-write. 1=Fatal error. 0=Non-fatal error.
3:0	Reserved.

**D2F[5:1]x160 Correctable Error Status**

Cold reset: 0000\_0000h.

Bits	Description
31:16	Reserved.
15	<b>HdrLogOvflStatus: header log overflow status.</b> Read-only.
14	<b>CorrIntErrStatus: corrected internal error status.</b> Read; Write-1-to-clear.
13	<b>AdvisoryNonfatalErrStatus: advisory non-fatal error status.</b> Read; Write-1-to-clear. 1=A non-fatal unsupported request errors or a non-fatal unexpected completion errors occurred.
12	<b>ReplayTimerTimeoutStatus: replay timer timeout status.</b> Read; Write-1-to-clear.
11:9	Reserved.
8	<b>ReplayNumRolloverStatus: replay.</b> 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	<b>BadDllpStatus: bad data link layer packet status.</b> Read; Write-1-to-clear. 1=A link CRC error was detected.
6	<b>BadTlpStatus: bad transaction layer packet status.</b> Read; Write-1-to-clear. 1=A bad non-duplicated sequence ID or a link CRC error was detected.
5:1	Reserved.
0	<b>RcvErrStatus: receiver error status.</b> Read-only. 1=An 8B10B or disparity error was detected.

### D2F[5:1]x164 Correctable Error Mask

Cold reset: 0000\_6000h.

Bits	Description
31:16	Reserved.
15	<b>HdrLogOvfMask: header log overflow mask.</b> Read-only.
14	<b>CorrIntErrMask: corrected internal error mask.</b> Read-write.
13	<b>AdvisoryNonfatalErrMask: advisory non-fatal error mask.</b> Read-write. 1=Error is not reported.
12	<b>ReplayTimerTimeoutMask: replay timer timeout mask.</b> Read-write. 1=Error is not reported.
11:9	Reserved.
8	<b>ReplayNumRolloverMask: replay.</b> Read-write. 1=Error is not reported.
7	<b>BadDllpMask: bad data link layer packet mask.</b> Read-write. 1=Error is not reported.
6	<b>BadTlpMask: bad transaction layer packet mask.</b> Read-write. 1=Error is not reported.
5:1	Reserved.
0	<b>RcvErrMask: receiver error mask.</b> Read-only. 1=Error is not reported.

### D2F[5:1]x168 Advanced Error Control

Cold reset: 0000\_0000h.

Bits	Description
31:12	Reserved.
11	<b>TlpPrefixLogPresent.</b> Read-only. IF (D2F[5:1]x7C[EndEndTlpPrefixSupported]==0) THEN Reserved. ENDIF. 1=If FirstErrPtr is valid then the TLP Prefix Log register contains valid information.
10	<b>MultiHdrRecdEn.</b> Read-only. 1=Enables recording more than one error header.
9	<b>MultiHdrRecdCap.</b> Read-only. 1=Specifies that the function is capable of recording more than one error header.
8	<b>EcrcCheckEn: data link protocol error severity.</b> Read-write. 0=Specifies that End-to-end CRC generation is not supported.
7	<b>EcrcCheckCap: data link protocol error severity.</b> Read-only. 0=Specifies that end-to-end CRC check is not supported.
6	<b>EcrcGenEn: end-to-end CRC enable.</b> Read-only. 0=Specifies that End-to-end CRC generation is not supported.

5	<b>EcreGenCap: end-to-end CRC capability.</b> Read-only. 0=Specifies that end-to-end CRC generation is not supported.
4:0	<b>FirstErrPtr: first error pointer.</b> Read-only. The First Error Pointer identifies the bit position of the first error reported in the Uncorrectable Error Status register.

**D2F[5:1]x16C Header Log DW0**

Cold reset: 0000\_0000h.

Bits	Description
31:0	<b>TlpHdr: transaction layer packet header log.</b> Read-only. Contains the header for a transaction layer packet corresponding to a detected error. The upper byte represents byte 0 of the header.

**D2F[5:1]x170 Header Log DW1**

Cold reset: 0000\_0000h.

Bits	Description
31:0	<b>TlpHdr: transaction layer packet header log.</b> Read-only. Contains the header for a transaction layer packet corresponding to a detected error. The upper byte represents byte 4 of the header.

**D2F[5:1]x174 Header Log DW2**

Cold reset: 0000\_0000h.

Bits	Description
31:0	<b>TlpHdr: transaction layer packet header log.</b> Read-only. Contains the header for a transaction layer packet corresponding to a detected error. The upper byte represents byte 8 of the header.

**D2F[5:1]x178 Header Log DW3**

Cold reset: 0000\_0000h.

Bits	Description
31:0	<b>TlpHdr: transaction layer packet header log.</b> Read-only. Contains the header for a transaction layer packet corresponding to a detected error. The upper byte represents byte 12 of the header.

**D2F[5:1]x17C Root Error Command**

Reset: 0000\_0000h.

Bits	Description
31:3	Reserved.
2	<b>FatalErrRepEn: fatal error reporting enable.</b> 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	<b>NonfatalErrRepEn: non-fatal error reporting enable.</b> 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	<b>CorrErrRepEn: correctable error reporting enable.</b> 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.

**D2F[5:1]x180 Root Error Status**

Cold reset: 0000\_0000h.

Bits	Description
31:27	<b>AdvErrIntMsgNum: advanced error interrupt message number.</b> Read-only.
26:7	Reserved.
6	<b>NFatalErrMsgRcvd: fatal error message received.</b> Read; Write-1-to-clear. Set to 1 when one or more fatal uncorrectable error messages have been received.
5	<b>NonFatalErrMsgRcvd: non-fatal error message received.</b> Read; Write-1-to-clear. Set to 1 when one or more non-fatal uncorrectable error messages have been received.
4	<b>FirstUncorrFatalRcvd: first uncorrectable fatal error message received.</b> Read; Write-1-to-clear. Set to 1 when the first uncorrectable error message received is for a fatal error.
3	<b>MultErrFatalNonfatalRcvd: ERR_FATAL/NONFATAL message received.</b> Read; Write-1-to-clear. Set when either a fatal or a non-fatal error is received and ErrFatalNonfatalRcvd is already set.
2	<b>ErrFatalNonfatalRcvd: ERR_FATAL/NONFATAL message received.</b> 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	<b>MultErrCorrRcvd: multiple ERR_COR messages received.</b> Read; Write-1-to-clear. Set when a correctable error message is received and ErrCorrRcvd is already set.
0	<b>ErrCorrRcvd: ERR_COR message received.</b> Read; Write-1-to-clear. Set when a correctable error message is received and this bit is not already set.

**D2F[5:1]x184 Error Source ID**

Cold reset: 0000\_0000h.

Bits	Description
31:16	<b>ErrFatalNonfatalSrcID: ERR_FATAL/ERR_NONFATAL source identification.</b> Read-only. Loaded with the requestor ID indicated in the received ERR_FATAL or ERR_NONFATAL message when D2F[5:1]x180[ErrFatalNonfatalRcvd] is not already set.
15:0	<b>ErrCorlSrcID: ERR_COR source identification.</b> Read-only. Loaded with the requestor ID indicated in the received ERR_COR message when D2F[5:1]x180[ErrCorrRcvd] is not already set.

### 3.9 Device 18h Function 0 Configuration Registers

See 3.1 [Register Descriptions and Mnemonics]. See 2.7 [Configuration Space].

#### D18F0x00 Device/Vendor ID

Bits	Description
31:16	<b>DeviceID: device ID.</b> Read-only. Value: 1530h.
15:0	<b>VendorID: vendor ID.</b> Read-only. Value: 1022h.

#### D18F0x04 Status/Command

Bits	Description
31:16	<b>Status.</b> Read-only. Value: 0010h. Bit[20] is set to indicate the existence of a PCI-defined capability block.
15:0	<b>Command.</b> Read-only. Value: 0000h.

#### D18F0x08 Class Code/Revision ID

Bits	Description
31:8	<b>ClassCode.</b> Read-only. Value: 060000h. Provides the host bridge class code as defined in the PCI specification.
7:0	<b>RevID: revision ID.</b> Read-only. Value: 00h.

#### D18F0x0C Header Type

Read-only. Value: 0080\_0000h.

Bits	Description
31:0	<b>HeaderTypeReg.</b> These bits are fixed at their default values. The header type field indicates that there are multiple functions present in this device.

#### D18F0x34 Capabilities Pointer

Bits	Description
31:8	Reserved.
7:0	<b>CapPtr: capabilities pointer.</b> Read-only. Value: 00h.

#### D18F0x[5C:40] Routing Table

Reset: 0004\_0201h.

**Table 101: Register Mapping** for D18F0x[5C:40]

Register	Function
D18F0x40	Node 0
D18F0x[5C:44]	Reserved

Bits	Description
31:0	Reserved.

**D18F0x60 Node ID**

Bits	Description										
31:21	Reserved.										
20:16	<b>CpuCnt[4:0]: CPU count bits[4:0].</b> Read-write. Reset: 0. Specifies the number of cores to be enabled (the boot core plus those cores enabled through D18F0x1DC[CpuEn]). <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00h</td> <td>1 core</td> </tr> <tr> <td>02h-01h</td> <td>&lt;CpuCnt[4:0] + 1&gt; cores</td> </tr> <tr> <td>03h</td> <td>4 cores</td> </tr> <tr> <td>1Fh-04h</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Description	00h	1 core	02h-01h	<CpuCnt[4:0] + 1> cores	03h	4 cores	1Fh-04h	Reserved
Bits	Description										
00h	1 core										
02h-01h	<CpuCnt[4:0] + 1> cores										
03h	4 cores										
1Fh-04h	Reserved										
15:0	Reserved.										

**D18F0x64 Unit ID**

Reset: 0000\_00E0h.

Bits	Description
31:16	Reserved.
15:11	Reserved.
10:8	Reserved.
7:6	<b>HbUnit: host bridge Unit ID.</b> Read-only. Specifies the coherent link Unit ID of the host bridge used by the coherent fabric.
5:4	<b>MctUnit: memory controller Unit ID.</b> Read-only. Specifies the coherent link Unit ID of the memory controller.
3:0	Reserved.

**D18F0x68 Link Transaction Control**

Bits	Description										
31	Reserved.										
30:28	Reserved.										
27:26	Reserved.										
25	Reserved.										
24	Reserved.										
23	<b>InstallStateS</b> . Read-write. Reset: 0. 1=Forces the default read block (RdBlk) install state to be shared instead of exclusive.										
22:21	<b>DsNpReqLmt: downstream non-posted request limit</b> . Read-write. Reset: 00b. BIOS: 10b. This specifies the maximum number of downstream non-posted requests issued by core(s) which may be outstanding on the IO links attached to this node at one time. <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>No limit</td> </tr> <tr> <td>01b</td> <td>limited to 1</td> </tr> <tr> <td>10b</td> <td>limited to 4</td> </tr> <tr> <td>11b</td> <td>limited to 8</td> </tr> </tbody> </table>	Bits	Description	00b	No limit	01b	limited to 1	10b	limited to 4	11b	limited to 8
Bits	Description										
00b	No limit										
01b	limited to 1										
10b	limited to 4										
11b	limited to 8										
20	<b>SeqIdSrcNodeEn: sequence ID source node enable</b> . Read-write. Reset: 0. 1=The source node ID of requests is provided in the SeqID field of the corresponding downstream IO link request packets. This may be useful for debug applications, in order to match downstream packets with their originating node. For normal operation, this bit should be cleared. Correct ordering of requests between different nodes is not ensured when this bit is set. Semaphore sharing between differing nodes may not work properly in systems which are capable of processing IO requests with differing non-zero SeqIDs out of request order.										
19	<b>ApicExtSpur: APIC extended spurious vector enable</b> . Read-write. Reset: 0. This enables the extended APIC spurious vector functionality; it affects APICF0[Vector]. 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	<b>ApicExtId: APIC extended ID enable</b> . Read-write. Reset: 0. Enables the extended APIC ID functionality. 0=APIC ID is 4 bits. 1=APIC ID is 8 bits.										
17	<b>ApicExtBrdCst: APIC extended broadcast enable</b> . Read-write. Reset: 0. Enables the extended APIC broadcast functionality. 0=APIC broadcast is 0Fh. 1=APIC broadcast is FFh. If ApicExt-BrdCst=1 then software must assert ApicExtId.										
16	<b>LintEn: local interrupt conversion enable</b> . 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. LINT0 and LINT1 are controlled by APIC3[60:50]. 0=ExtInt/NMI interrupts delivered unchanged.										
15	<b>LimitCldtCfg: limit coherent link configuration space range</b> . Read-write. Reset: 0. BIOS: 1.										
14:13	Reserved.										



12	Reserved.
11	<b>RespPassPW: response PassPW.</b> Read-write. Reset: 0. BIOS: 1. 1=The PassPW bit in all downstream link 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	<b>DisFillP: disable fill probe.</b> Read-write. Reset: 0. Controls probes for core-generated fills. 0=Probes issued for cache fills. 1=Probes not issued for cache fills. BIOS: 0. BIOS may set if single core.
9	<b>DisRmtPMemC: disable remote probe memory cancel.</b> Read-write. Reset: 0. 1=Only probed caches on the same node as the target memory controller may generate MemCancel coherent link packets. MemCancels are used to attempt to save DRAM and/or link bandwidth associated with the transfer of stale DRAM data. 0=Probes hitting dirty blocks may generate MemCancel packets, regardless of the location of the probed cache.
8	<b>DisPMemC: disable probe memory cancel.</b> Read-write. Reset: 0. Controls generation of MemCancel coherent link packets. MemCancels are used to attempt to save DRAM and/or coherent link bandwidth associated with the transfer of stale DRAM data. 0=Probes hitting dirty blocks of the core cache may generate MemCancel packets. 1=Probes may not generate MemCancel packets.
7	<b>CPURdRspPassPW: CPU read response PassPW.</b> 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	<b>CPUReqPassPW: CPU request PassPW.</b> 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	Reserved.
4	<b>DisMTS: disable memory controller target start.</b> Read-write. Reset: 0. BIOS: 1. 1=Disables use of TgtStart. TgtStart is used to improve scheduling of back-to-back ordered transactions by indicating when the first transaction is received and ordered at the memory controller.
3	<b>DisWrDwP: disable write doubleword probes.</b> Read-write. Reset: 0. BIOS: 1. 1=Disables generation of probes for core-generated, WrSized doubleword commands.
2	<b>DisWrBP: disable write byte probes.</b> Read-write. Reset: 0. BIOS: 1. 1=Disables generation of probes for core-generated, WrSized byte commands.
1	<b>DisRdDwP: disable read doubleword probe.</b> Read-write. Reset: 0. BIOS: 1. 1=Disables generation of probes for core-generated, RdSized doubleword commands.
0	<b>DisRdBp: disable read byte probe.</b> Read-write. Reset: 0. BIOS: 1. 1=Disables generation of probes for core-generated, RdSized byte commands.

### D18F0x6C Link Initialization Control

Bits	Description
31	Reserved.

30	<b>RlsLnkFullTokCntImm: release upstream full token count immediately.</b> Read-write. Reset: 0b. BIOS: 1 after buffer counts have been programmed. 1=Apply buffer counts programmed in <a href="#">D18F0x[F0,D0,B0,90]</a> and <a href="#">D18F0x[F4,D4,B4,94]</a> immediately without requiring warm reset. Once this bit is set, additional changes to the buffer counts only take effect upon warm reset.
29	Reserved.
28	<b>RlsIntFullTokCntImm: release internal full token count immediately.</b> Read-write. Reset: 0b. BIOS: 1 after buffer counts have been programmed. 1=Apply buffer counts programmed in <a href="#">D18F3x6C</a> , <a href="#">D18F3x70</a> , <a href="#">D18F3x74</a> , <a href="#">D18F3x78</a> , <a href="#">D18F3x7C</a> , <a href="#">D18F3x140</a> , <a href="#">D18F3x144</a> , <a href="#">D18F3x1[54:48]</a> , <a href="#">D18F3x17C</a> , and <a href="#">D18F3x1A0</a> immediately without requiring warm reset. Once this bit is set, additional changes to the buffer counts only take effect upon warm reset.
27	<b>ApplyIsocModeEnNow.</b> Read-write. Reset: 0b. BIOS: 1 after RlsLnkFullTokCntImm and RlsIntFullTokCntImm have been set. 1=Apply the programmed value in <a href="#">D18F0x[E4,C4,A4,84][IsocEn]</a> immediately without requiring warm reset. This bit may only be set if RlsLnkFullTokCntImm and RlsIntFullTokCntImm are set and isochronous buffers have been allocated. IF (ApplyIsocModeEnNow) THEN ( <a href="#">D18F3x1[54:48][IsocPreqTok0]</a> > 0).
26:24	Reserved.
23	Reserved.
22:21	Reserved.
20	Reserved.
19:16	Reserved.
15:12	Reserved.
11	Reserved.
10:9	<b>BiosRstDet[2:1]: BIOS reset detect bits[2:1].</b> See: BiosRstDet[0].
8	Reserved.
7	Reserved.
6	<b>InitDet: CPU initialization command detect.</b> 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	<b>BiosRstDet[0]: BIOS reset detect bit[0].</b> Read-write. Cold reset: 0. BiosRstDet[2:0] = {BiosRstDet[2:1], BiosRstDet[0]}. May be used to distinguish between a reset event generated by the 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	<b>ColdRstDet: cold reset detect.</b> 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.
3:2	Reserved.

1	Reserved.
0	<b>RouteTblDis: routing table disable.</b> Read-write. Reset: 1. BIOS: 0.

### **D18F0x[E4,C4,A4,84] Link Control**

Table 102: Register Mapping for D18F0x[E4,C4,A4,84]

Register	Function
D18F0x84	ONION Link
D18F0x[E4,C4,A4]	Reserved

This register is derived from the link control register defined in the link specification.

Bits	Description
31:16	Reserved.
15	<b>Addr64BitEn: 64-bit address packet enable.</b> Read-write. Reset: 0000b. 1=Requests to addresses greater than FF_FFFF_FFFFh are supported by this IO link. 0=Requests to addresses greater than FF_FFFF_FFFFh are master aborted as if the end of chain was reached. BIOS is required to ensure that the link-specification-defined “64 Bit Address Feature” bit in the device on the other side of the link is set prior to setting this bit. For coherent links, this bit is unused. <a href="#">D18F0x68[CHtExtAddrEn]</a> is required to be set if this bit is set for any IO link. The link specification indicates that this bit is cleared by a warm reset; therefore this bit may be in a different state than an IO device on the other side of the link after a warm reset; care should be taken by BIOS to place devices on both sides of the link in the same state after a warm reset, before any packets to the high-order addresses enabled by this bit are generated.
14:13	Reserved.
12	<b>IsocEn: isochronous flow-control mode enable.</b> Read-write. Reset: 0b. BIOS: 1 if the link is an ONION Link. This bit is set to place the link into isochronous flow-control mode (IFCM), as defined by the link specification. 1=IFCM. 0=Normal flow-control mode. See <a href="#">D18F0x6C[ApplyIsocModeEnNow]</a> .
11:6	Reserved.
5	Reserved.
4	<b>LinkFail: link failure.</b> Read; set-by-hardware; write-1-to-clear. Cold reset: 0. This bit is set high by the hardware if a sync flood is received by the link. See <a href="#">2.14.1.9.1 [Common Diagnosis Information]</a> .
3:0	Reserved.

### **D18F0x[EC,CC,AC,8C] Link Feature Capability**

This register is derived from the link feature capability register defined in the link specification. Unless otherwise specified: 0=The feature is not supported; 1=The feature is supported.

Table 103: Register Mapping for D18F0x[EC,CC,AC,8C]

Register	Function
D18F0x8C	ONION Link
D18F0x[EC,CC,AC]	Reserved

Bits	Description
31:6	Reserved.
5	<b>UnitIdReOrderDis: UnitID reorder disable.</b> Read-write. Reset: 0. 1=Upstream reordering for different UnitIDs is not supported; i.e., all upstream packets are ordered as if they have the same UnitID. 0=Reordering based on UnitID is supported.
4	Reserved. Reset: 1.
3:2	Reserved.
1	Reserved. Reset: 1.
0	Reserved. Reset: 1.

#### **D18F0x[F0,D0,B0,90] Link Base Channel Buffer Count**

Read-write; Reset-applied.

Table 104: Register Mapping for D18F0x[F0,D0,B0,90]

Register	Function
D18F0x90	ONION Link
D18F0x[F0,D0,B0]	Reserved

D18F0x[F0,D0,B0,90] and D18F0x[F4,D4,B4,94] specify the *hard-allocated* link flow-control buffer counts in each virtual channel available to the transmitter at the other end of the link; it also provides the *free buffers* that may be used by any of the virtual channels, as needed. Base channel buffers are specified in D18F0x[F0,D0,B0,90]; isochronous buffer counts (if in IFCM) are specified in D18F0x[F4,D4,B4,94]. For all fields that specify buffer counts in D18F0x[F0,D0,B0,90] and D18F0x[F4,D4,B4,94], if the link is ganged, then the number of buffers allocated is 2 times the value of the field; If the link is unganged, then the number of buffers allocated is the value of the field.

The cold or warm reset value is determined by whether the link initializes, whether the link is IO/coherent, whether the link is ganged/unganged, and whether the settings are locked by LockBc. Out of cold reset, the processor allocates a minimal number of buffers that is smaller than the default values in the register. BIOS must use D18F0x6C[RlsLnkFullTokCntImm] for the values in the register to take effect. This is necessary even if the values are unchanged from the default values.

The hard-allocated buffer counts are transmitted to the device at the other end of the link in buffer release messages after link initialization. The remaining buffers are held in the free list (specified by FreeData and FreeCmd) used to optimize buffer usage. When a transaction is received, if a free-list buffer is available, it is used for storage instead of one of the hard allocated buffers; as a result, a buffer release (for one of the hard allocated buffers used by the incoming request) can be immediately sent back to the device at the other end of the link without waiting for the transaction to be routed beyond the flow-control buffers.

**Table 105: Link Buffer Definitions**

Term	Definition
<b>LpbSize</b>	Link Packet Command Buffer size. LpbSize = 24.
<b>LpbdSize</b>	Link Packet Data Buffer size. LpbdSize = 16.
<b>LcsSize</b>	Link Command Scheduler size. LcsSize = 32.

Buffer allocation rules:

- The total number of command buffers allocated in the base and isochronous registers of a link cannot exceed **LpbSize**:
  - $(D18F0x[F0,D0,B0,90][NpReqCmd] + D18F0x[F0,D0,B0,90][PReq] + D18F0x[F0,D0,B0,90][RspCmd] + D18F0x[F0,D0,B0,90][ProbeCmd] + D18F0x[F0,D0,B0,90][FreeCmd] + D18F0x[F4,D4,B4,94][IsocNpReqCmd] + D18F0x[F4,D4,B4,94][IsocPReq] + D18F0x[F4,D4,B4,94][IsocRspCmd]) \leq LpbSize.$
- The total number of data buffers allocated in the base and isochronous registers of a link cannot exceed **LpbdSize**:
  - $(D18F0x[F0,D0,B0,90][NpReqData] + D18F0x[F0,D0,B0,90][RspData] + D18F0x[F0,D0,B0,90][PReq] + D18F0x[F0,D0,B0,90][FreeData] + D18F0x[F4,D4,B4,94][IsocPReq] + D18F0x[F4,D4,B4,94][IsocNpReqData] + D18F0x[F4,D4,B4,94][IsocRspData]) \leq LpbdSize.$
- The total number of hard allocated command buffers cannot exceed **LcsSize**.
  - $(D18F0x[F0,D0,B0,90][ProbeCmd] + D18F0x[F0,D0,B0,90][RspCmd] + D18F0x[F0,D0,B0,90][PReq] + D18F0x[F0,D0,B0,90][NpReqCmd] + D18F0x[F4,D4,B4,94][IsocRspCmd] + D18F0x[F4,D4,B4,94][IsocPReq] + D18F0x[F4,D4,B4,94][IsocNpReqCmd]) \leq LcsSize.$

Bits	Description
31	<b>LockBc: lock buffer count register.</b> Cold reset: 0. BIOS: 1. 1=The buffer count registers, <a href="#">D18F0x[F0,D0,B0,90]</a> and <a href="#">D18F0x[F4,D4,B4,94]</a> are locked such that warm resets do not place the registers back to their default value. Setting this bit does not prevent the buffer counts from being updated after a warm reset based on the value of the buffer counts before the warm reset. 0=Upon warm reset, the buffer count registers return to their default value after the link initializes regardless of the value before the warm reset.
30	<b>PReq[3]: posted request command and data buffer count [3].</b> IF (LockBc) THEN Cold reset: 0. ELSE Reset: 0. ENDIF. BIOS: 0. See: PReq[2:0].
29:28	<b>NpReqData[3:2]: non-posted request data buffer count [3:2].</b> IF (LockBc) THEN Cold reset: 00b. ELSE THEN Reset: 00b. ENDIF. BIOS: IF (REG==D18F0x90) THEN 00b ELSE 11b ENDIF. See: NpReqData[1:0].

27:25	<b>FreeData: free data buffer count.</b> IF (D18F0x[F0,D0,B0,90][LockBc]) THEN Cold reset: 0. ELSE THEN Reset: 0. ENDIF. BIOS: 0.
24:20	<b>FreeCmd: free command buffer count.</b> IF (D18F0x[F0,D0,B0,90][LockBc]) THEN Cold reset: 0. ELSE THEN Reset: 0. ENDIF. BIOS: IF (REG==D18F0x90) THEN 00h ELSE 00h ENDIF.
19:18	<b>RspData: response data buffer count.</b> IF (LockBc) THEN Cold reset: 1. ELSE THEN Reset: 1. ENDIF. BIOS: IF (REG==D18F0x90) THEN 1 ELSE 0 ENDIF.
17:16	<b>NpReqData[1:0]: non-posted request data buffer count [1:0].</b> NpReqData[3:0] = {NpReqData[3:2], NpReqData[1:0]}. IF (LockBc) THEN Cold reset: 01b. ELSE THEN Reset: 01b. ENDIF. BIOS: IF (REG==D18F0x90) THEN 01b ELSE 11b ENDIF.
15:12	<b>ProbeCmd: probe command buffer count.</b> IF (LockBc) THEN Cold reset: 0h. ELSE THEN Reset: 0h. ENDIF. BIOS: 0h.
11:8	<b>RspCmd: response command buffer count.</b> IF (LockBc) THEN Cold reset: 1h. ELSE THEN Reset: 1h. ENDIF. BIOS: IF (REG==D18F0x90) THEN 1h ELSE 0h ENDIF.
7:5	<b>PReq[2:0]: posted request command and data buffer count [2:0].</b> PReq[3:0] = {PReq[3], PReq[2:0]}. Specifies the number of posted command and posted data buffers allocated. IF (LockBc) THEN Cold reset: 110b. ELSE Reset: 110b. ENDIF. BIOS: 101b.
4:0	<b>NpReqCmd: non-posted request command buffer count.</b> IF (LockBc) THEN Cold reset: 09h. ELSE THEN Reset: 09h. ENDIF. BIOS: IF (REG==D18F0x90) THEN 05h ELSE 18h ENDIF.

### **D18F0x[F4,D4,B4,94] Link Isochronous Channel Buffer Count**

Read-write; Reset-applied. See [D18F0x\[F0,D0,B0,90\]](#).

Table 106: [Register Mapping](#) for [D18F0x\[F4,D4,B4,94\]](#)

Register	Function
D18F0x94	ONION Link
D18F0x[F4,D4,B4]	Reserved

Bits	Description
31:29	Reserved.

28:27	<b>IsocRspData: isochronous response data buffer count.</b> IF (D18F0x[F0,D0,B0,90][LockBc]) THEN Cold reset: 0. ELSE THEN Reset: 0. ENDIF. BIOS: 0.
26:25	<b>IsocNpReqData: isochronous non-posted request data buffer count.</b> IF (D18F0x[F0,D0,B0,90][LockBc]) THEN Cold reset: 0. ELSE THEN Reset: 0. ENDIF. BIOS: IF (REG==D18F0x94) THEN 1 ELSE 0 ENDIF.
24:22	<b>IsocRspCmd: isochronous response command buffer count.</b> IF (D18F0x[F0,D0,B0,90][LockBc]) THEN Cold reset: 0. ELSE THEN Reset: 0. ENDIF. BIOS: 0.
21:19	<b>IsocPReq: isochronous posted request command and data buffer count.</b> IF (D18F0x[F0,D0,B0,90][LockBc]) THEN Cold reset: 0. ELSE THEN Reset: 0. ENDIF. This specifies the number of isochronous posted command and posted data buffers allocated. BIOS: 0.
18:16	<b>IsocNpReqCmd: isochronous non-posted request command buffer count.</b> IF (D18F0x[F0,D0,B0,90][LockBc]) THEN Cold reset: 0. ELSE THEN Reset: 0. ENDIF. BIOS: IF (REG==D18F0x94) THEN 1 ELSE 0 ENDIF.
15:8	<b>SecBusNum: secondary bus number.</b> Reset: 0. Specifies the configuration-space bus number of the IO link. When configured as a coherent link, this register has no meaning. This field should match the corresponding D18F1x[1DC:1D0,EC:E0][BusNumBase], unless D18F1x[1DC:1D0,EC:E0][DevCmpEn]=1, in which case this field should be 00h).
7:0	Reserved.

**D18F0x[F8,D8,B8,98] Link Type**

Table 107: Register Mapping for D18F0x[F8,D8,B8,98]

Register	Function
D18F0x98	ONION Link
D18F0x[F8,D8,B8]	Reserved

Bits	Description
31:6	Reserved.
5	<b>PciEligible.</b> Read-only. Reset: 1.
4:3	Reserved.
2	Reserved.
1	Reserved.
0	Reserved.

**D18F0x[11C,118,114,110] Link Clumping Enable**

Reset: 0000\_0000h. D18F0x[11C,118,114,110] are associated with the whole link if it is ganged or sublink 0 if it is unganged; If the node does not support a link, then the corresponding register addresses become reserved.

Table 108: Register Mapping for D18F0x[11C,118,114,110]

Register	Function
D18F0x110	ONION Link
D18F0x11[C:4]	Reserved

These registers specify how UnitIDs of upstream non-posted requests may be clumped per the link specification. The processor does not clump requests that it generates in the downstream direction.

Bits	Description
31:1	<b>ClumpEn.</b> Read-write. Each bit of this register corresponds to a link UnitID number. E.g., bit 2 corresponds to UnitID 02h, etc. 1=The specified UnitID is ordered in the same group as the specified UnitID - 1. For example if this register is programmed to 0000_00C0h, then UnitIDs 7h, 6h, and 5h are all ordered as if they are part of the same UnitID. This is used to allow more than 32 tags to be assigned to a single stream for the purposes of ordering.
0	Reserved.

#### D18F0x150 Link Global Retry Control

Cold reset: 0000\_0000h.

Bits	Description
31:0	Reserved.

#### D18F0x168 Extended Link Transaction Control

Read-write.

Bits	Description
31:21	Reserved.
20	<b>XcsSecPickerDstNcHt.</b> Reset: 0. BIOS: 1.
19	Reserved.
18	Reserved.
17:15	Reserved.
14:12	Reserved.
11:0	Reserved.

#### D18F0x16C Link Global Extended Control

Reset: 0000\_0000h.

Bits	Description
31:0	Reserved.



**D18F0x[18C:170] Link Extended Control**

These registers provide control for each link. They are mapped to the links as follows:

Table 109: Register Mapping for D18F0x[18C:170]

Register	Function
D18F0x170	ONION Link
D18F0x1[8C:74]	Reserved

Reset: 0000\_0001h.

Bits	Description
31:0	Reserved.

**D18F0x1A0 Link Initialization Status**

Table 110: Onion Definitions

Term	Definition
<b>OnionPlus</b>	OnionPlus link detected. OnionPlus = (D18F0x1A0[OnionPlusCap]).

Bits	Description								
31	<b>InitStatusValid: initialization status valid.</b> Read-only; Updated-by-hardware. Reset: 0. 1=Indicates that the rest of the information in this register is valid for all links; each link is either not connected or the initialization is complete.								
30:28	Reserved.								
27:24	<b>OnionPlusCap.</b> Read-only; Updated-by-hardware. Reset: 0h. 1=OnionPlus capable link detected. <table border="1"> <thead> <tr> <th>Bit</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>[0]</td> <td>Link 0</td> </tr> <tr> <td>[1]</td> <td>Link 1</td> </tr> <tr> <td>[2:3]</td> <td>Reserved</td> </tr> </tbody> </table>	Bit	Description	[0]	Link 0	[1]	Link 1	[2:3]	Reserved
Bit	Description								
[0]	Link 0								
[1]	Link 1								
[2:3]	Reserved								
23:4	Reserved.								
3:2	<b>InitComplete1: initialization complete for link 1.</b> See: InitComplete0.								
1:0	<b>InitComplete0: initialization complete for link 0.</b> Read-only; Updated-by-hardware. Reset: 00b. <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Internal northbridge link has not completed initialization.</td> </tr> <tr> <td>10b-01b</td> <td>Reserved</td> </tr> <tr> <td>11b</td> <td>Internal northbridge link has completed initialization.</td> </tr> </tbody> </table>	Bits	Description	00b	Internal northbridge link has not completed initialization.	10b-01b	Reserved	11b	Internal northbridge link has completed initialization.
Bits	Description								
00b	Internal northbridge link has not completed initialization.								
10b-01b	Reserved								
11b	Internal northbridge link has completed initialization.								

**D18F0x1DC Core Enable**

Reset: 0000\_0000h.

Bits	Description
31:16	Reserved.

15:1	<p><b>CpuEn: core enable.</b> Read-write. This field is used to enable each of the cores after a reset. 1=Enable the core to start fetching and executing code from the boot vector. The most significant bit N is indicated by CpuCoreNum, as defined in section 2.4.4 [Processor Cores and Downcoring]. All bits greater than N are reserved.</p> <table border="1"> <thead> <tr> <th data-bbox="277 338 321 365"><u>Bit</u></th> <th data-bbox="521 338 662 365"><u>Description</u></th> </tr> </thead> <tbody> <tr> <td data-bbox="277 371 321 399">[0]</td> <td data-bbox="521 371 683 399">Core 1 enable</td> </tr> <tr> <td data-bbox="277 405 321 432">[1]</td> <td data-bbox="521 405 781 432">Core &lt;BIT+1&gt; enable</td> </tr> <tr> <td data-bbox="277 438 321 466">[2]</td> <td data-bbox="521 438 683 466">Core 3 enable</td> </tr> <tr> <td data-bbox="277 472 354 499">[15:3]</td> <td data-bbox="521 472 634 499">Reserved</td> </tr> </tbody> </table>	<u>Bit</u>	<u>Description</u>	[0]	Core 1 enable	[1]	Core <BIT+1> enable	[2]	Core 3 enable	[15:3]	Reserved
<u>Bit</u>	<u>Description</u>										
[0]	Core 1 enable										
[1]	Core <BIT+1> enable										
[2]	Core 3 enable										
[15:3]	Reserved										
0	Reserved.										

### 3.10 Device 18h Function 1 Configuration Registers

See 3.1 [Register Descriptions and Mnemonics]. See 2.7 [Configuration Space].

#### D18F1x00 Device/Vendor ID

Bits	Description
31:16	<b>DeviceID: device ID.</b> Read-only. Value: 1531h.
15:0	<b>VendorID: vendor ID.</b> Read-only. Value: 1022h.

#### D18F1x08 Class Code/Revision ID

Bits	Description
31:8	<b>ClassCode.</b> Read-only. Value: 060000h. Provides the host bridge class code as defined in the PCI specification.
7:0	<b>RevID: revision ID.</b> Read-only. Value: 00h. Processor revision. 00h=A0.

#### D18F1x0C Header Type

Reset: 0080\_0000h.

Bits	Description
31:0	<b>HeaderTypeReg.</b> Read-only. These bits are fixed at their default values. The header type field indicates that there are multiple functions present in this device.

#### D18F1x[17C:140,7C:40] DRAM Base/Limit

The following sets of registers specify the DRAM address ranges:

Table 111: Register Mapping for D18F1x[17C:140,7C:40]

Function	Base Low	Limit Low	Base High	Limit High
Range 0	D18F1x40	D18F1x44	D18F1x140	D18F1x144
Reserved	D18F1x48, D18F1x[7:5][8,0]	D18F1x4C, D18F1x[7:5][C,4]	D18F1x148, D18F1x1[7:5][8,0]	D18F1x14C, D18F1x1[7:5][C,4]

Transaction addresses that are within the specified base/limit range are routed to the DstNode. See 2.8.2 [NB Routing].

DRAM mapping rules:

F1x0XX registers provide the low address bits. F1x1XX registers are reserved.

- Transaction addresses are within the defined range if:  
{DramBase[39:24], 00\_0000h} <= address[39:0] <= {DramLimit[39:24], FF\_FFFFh}.
- DRAM regions must not overlap each other.
- Accesses to addresses that map to both DRAM, as specified by the DRAM base and limit registers (F1x[1, 0][7C:40]), and MMIO, as specified by D18F1x[2CC:2A0,1CC:180,BC: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 IO links are routed based on the DRAM base and limit registers (F1x[1, 0][7C:40]) only.
- The appropriate RE or WE bit(s) must be set. When initializing a base/limit pair, the BIOS must write the [limit] register 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 [2.8.2.1.1 \[DRAM and MMIO Memory Space\]](#).

When memory hoisting is enabled in a node via [D18F1x2\[1,0\]\[8,0\]\[LgcyMmioHoleEn\]](#), the corresponding BaseAddr/LimitAddr should be configured to account for the memory hoisted above the hole. See [2.9.10 \[Memory Hoisting\]](#).

**D18F1x[7:4][8,0] DRAM Base Low**

IF ([D18F2x118\[LockDramCfg\]](#)) THEN Read-only. ELSE Read-write. ENDIF. Reset: 0000\_0000h.

Table 112: [Register Mapping](#) for [D18F1x\[7:4\]\[8,0\]](#)

Register	Function
D18F1x40	Range 0
D18F1x48	Reserved
D18F1x[7:5][8,0]	Reserved

Bits	Description
31:16	<b>DramBase[39:24]: DRAM base address register bits[39:24].</b>
15:2	Reserved.
1	<b>WE: write enable.</b> 1=Writes to this address range are enabled.
0	<b>RE: read enable.</b> 1=Reads to this address range are enabled.

**D18F1x[7:4][C,4] DRAM Limit Low**

IF ([D18F2x118\[LockDramCfg\]](#)) THEN Read-only. ELSE Read-write. ENDIF.

Table 113: [Register Mapping](#) for [D18F1x\[7:4\]\[C,4\]](#)

Register	Function
D18F1x44	Range 0
D18F1x4C	Reserved
D18F1x[7:5][C,4]	Reserved

Bits	Description
31:16	<b>DramLimit[39:24]: DRAM limit address register bits[39:24].</b> Reset: FCFFh.
15:11	Reserved.

10:8	Reserved.
7:3	Reserved.
2:0	<b>DstNode: destination Node ID.</b> Reset: 000b. Specifies the node that a packet is routed to if it is within the address range.

### **D18F1x[2CC:2A0,1CC:180,BC:80] MMIO Base/Limit**

These registers, The memory mapped IO base and limit registers **D18F1x[2CC:2A0,1CC:180,BC:80]** specify the mapping from memory addresses to the corresponding node and IO link for MMIO transactions. Address ranges are specified by upto 16 sets of base/limit registers.

Table 114: Register Mapping for **D18F1x[2CC:2A0,1CC:180,BC:80]**

Function	MMIO Base Low	MMIO Limit Low	MMIO Base/Limit High
Range 0	D18F1x80	D18F1x84	Reserved
Range 1	D18F1x88	D18F1x8C	Reserved
Range 2	D18F1x90	D18F1x94	Reserved
Range 3	D18F1x98	D18F1x9C	Reserved
Range 4	D18F1xA0	D18F1xA4	Reserved
Range 5	D18F1xA8	D18F1xAC	Reserved
Range 6	D18F1xB0	D18F1xB4	Reserved
Range 7	D18F1xB8	D18F1xBC	Reserved
Range 8	D18F1x1A0	D18F1x1A4	Reserved
Range 9	D18F1x1A8	D18F1x1AC	Reserved
Range 10	D18F1x1B0	D18F1x1B4	Reserved
Range 11	D18F1x1B8	D18F1x1BC	Reserved
Reserved	D18F1x2[B8,B0,A8,A0]	D18F1x2[BC,B4,AC,A4]	Reserved

Transaction addresses that are within the specified base/limit range are routed to the node specified by DstNode and the link specified by DstLink. See 2.8.2 [NB Routing].

#### MMIO mapping rules:

- Transaction addresses are within the defined range if:  
 $\{MMIOBase[39:16], 0000h\} \leq \text{address}[39:0] \leq \{MMIOLimit[39:16], FFFFh\}$ .
- MMIO regions must not overlap each other.
- Accesses to addresses that map to both DRAM, as specified by the DRAM base and limit registers (see **D18F1x[17C:140,7C:40]**), and MMIO, as specified by the memory mapped IO base and limit registers (**F1x[BC: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 IO links are routed based on **D18F1x[2CC:2A0,1CC:180,BC:80]**.
- The appropriate RE or WE bit(s) must be set. When initializing a base/limit pair, the BIOS must write the limit register 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.
- Scenarios in which the address space of multiple MMIO ranges target the same IO device is supported.

- See 2.8.2.1.1 [DRAM and MMIO Memory Space].

### D18F1x[2B:1A,B:8][8,0] MMIO Base Low

Table 115: Register Mapping for D18F1x[2B:1A,B:8][8,0]

Register	Function
D18F1x80	Range 0
D18F1x88	Range 1
D18F1x90	Range 2
D18F1x98	Range 3
D18F1xA0	Range 4
D18F1xA8	Range 5
D18F1xB0	Range 6
D18F1xB8	Range 7
D18F1x1A0	Range 8
D18F1x1A8	Range 9
D18F1x1B0	Range 10
D18F1x1B8	Range 11
D18F1x2[B:A][8,0]	Reserved

Bits	Description
31:8	<b>MMIOBase[39:16]: MMIO base address register bits[39:16].</b> Read-write. Reset: 0.
7:4	Reserved.
3	<b>Lock.</b> Read-write. Reset: 0. 1=the memory mapped IO base and limit registers (D18F1x[2CC:2A0,1CC:180,BC:80]) 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	<b>WE: write enable.</b> Read-write. Reset: 0. 1=Writes to this address range are enabled.
0	<b>RE: read enable.</b> Read-write. Reset: 0. 1=Reads to this address range are enabled.

### D18F1x[2B:1A,B:8][C,4] MMIO Limit Low

Table 116: Register Mapping for D18F1x[2B:1A,B:8][C,4]

Register	Function
D18F1x84	Range 0
D18F1x8C	Range 1
D18F1x94	Range 2
D18F1x9C	Range 3
D18F1xA4	Range 4
D18F1xAC	Range 5

Table 116: Register Mapping for D18F1x[2B:1A,B:8][C,4]

D18F1xB4	Range 6
D18F1xBC	Range 7
D18F1x1A4	Range 8
D18F1x1AC	Range 9
D18F1x1B4	Range 10
D18F1x1BC	Range 11
D18F1x2[BC,B4,AC, A4]	Reserved.

Bits	Description										
31:8	<b>MMIOLimit[39:16]: MMIO limit address register bits[39:16].</b> Read-write. Reset: 0.										
7	<p><b>NP: non-posted.</b> Read-write. Reset: 0. 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 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 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 IO links (the virtual channel of the request is specified by the IO request).</p> <p>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.</p>										
6	<b>DstSubLink: destination sublink.</b> Read-write. Reset: 0. When a link is unganged, this bit specifies the destination sublink of the link specified by the memory mapped IO base and limit registers F1x[BC:80][DstLink]. 0=The destination link is sublink 0. 1=The destination link is sublink 1. If the link is ganged, then this bit must be low.										
5:4	<p><b>DstLink: destination link ID.</b> Read-write. Reset: 0. For transactions within this MMIO range, this field specifies the destination IO link number of the destination node.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Link 0</td> </tr> <tr> <td>01b</td> <td>Link 1</td> </tr> <tr> <td>10b</td> <td>Link 2</td> </tr> <tr> <td>11b</td> <td>Link 3</td> </tr> </tbody> </table>	Bits	Description	00b	Link 0	01b	Link 1	10b	Link 2	11b	Link 3
Bits	Description										
00b	Link 0										
01b	Link 1										
10b	Link 2										
11b	Link 3										
3	Reserved.										
2:0	<b>DstNode: destination node ID bits.</b> Read-write. Reset: 0. For transactions within this MMIO range, this field specifies the destination node ID.										

**D18F1x[DC:C0] IO-Space Base/Limit**

The IO-space base and limit registers, D18F1x[DC:C0], specify the mapping from IO addresses to the

corresponding node and IO link for transactions resulting from x86-defined IN and OUT instructions. IO address ranges are specified by upto 8 sets of base/limit registers. The first set is F1xC0 and F1xC4, the second set is F1xC8 and F1xCC, and so forth. Transaction addresses that are within the specified base/limit range are routed to the node specified by DstNode and the link specified by DstLink. See 2.8.2 [NB Routing].

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.
- IO regions must not overlap each other.
- The appropriate RE or WE bit(s) must be set.
- See 2.8.2.1.2 [IO Space].

### D18F1x[1F:1E,D:C][8,0] IO-Space Base

Table 117: Register Mapping for D18F1x[1F:1E,D:C][8,0]

Register	Function
D18F1xC0	Range 0
D18F1xC8	Range 1
D18F1xD0	Range 2
D18F1xD8	Range 3
D18F1x1[F:E][8,0]	Reserved

Bits	Description
31:25	Reserved.
24:12	<b>IOBase[24:12]: IO base address register bits[24:12].</b> Read-write. Reset: 0.
11:6	Reserved.
5	<b>IE: ISA enable.</b> Read-write. Reset: 0. 1=The IO-space address window is limited to the first 256 B of each 1 KB block specified; this only applies to the first 64 KB of IO space. 0=The PCI IO window is not limited in this way.
4	<b>VE: VGA enable.</b> Read-write. Reset: 0. 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 64 KB 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. This bit should only ever be set in one register. 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 an IO link, see D18F1xF4 [VGA Enable]. When D18F1xF4[VE] is set, the state of this bit is ignored.
3:2	Reserved.
1	<b>WE: write enable.</b> Read-write. Reset: 0. 1=Writes to this IO-space address range are enabled.
0	<b>RE: read enable.</b> Read-write. Reset: 0. 1=Reads to this IO-space address range are enabled.



**D18F1x[1F:1E,D:C][C,4] IO-Space Limit**

Table 118: Register Mapping for D18F1x[1F:1E,D:C][C,4]

Register	Function
D18F1xC4	Range 0
D18F1xCC	Range 1
D18F1xD4	Range 2
D18F1xDC	Range 3
D18F1x1[F:E][C,4]	Reserved

Bits	Description										
31:25	Reserved.										
24:12	<b>IOLimit[24:12]: IO limit address register bits[24:12].</b> Read-write. Reset: 0.										
11:7	Reserved.										
6	<b>DstSubLink: destination sublink.</b> Read-write. Reset: 0. When a link is unganged, this bit specifies the destination sublink of the link specified by F1x[DC:C0][DstLink]. 0=The destination link is sublink 0. 1=The destination link is sublink 1. If the link is ganged, then this bit must be low.										
5:4	<b>DstLink: destination link ID.</b> Read-write. Reset: 0. For transactions within this IO-space range, this field specifies the destination IO link number of the destination node. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Link 0</td> </tr> <tr> <td>01b</td> <td>Link 1</td> </tr> <tr> <td>10b</td> <td>Link 2</td> </tr> <tr> <td>11b</td> <td>Link 3</td> </tr> </tbody> </table>	Bits	Description	00b	Link 0	01b	Link 1	10b	Link 2	11b	Link 3
Bits	Description										
00b	Link 0										
01b	Link 1										
10b	Link 2										
11b	Link 3										
3	Reserved.										
2:0	<b>DstNode: destination node ID bits.</b> Read-write. Reset: 0. For transactions within this IO-space range, this field specifies the destination node ID.										

**D18F1x[1DC:1D0,EC:E0] Configuration Map**

D18F1x[1DC:1D0,EC:E0] specify the mapping from configuration address to the corresponding node and IO link. Configuration address ranges are specified by upto 8 pairs of base/limit registers. Transaction addresses that are within the specified base/limit range are routed to the node specified by DstNode and the link specified by DstLink. See 2.8.2 [NB Routing].

Table 119: Register Mapping for D18F1x[1DC:1D0,EC:E0]

Register	Function
D18F1xE0	Range 0
D18F1xE4	Range 1
D18F1xE8	Range 2
D18F1xEC	Range 3
D18F1x1[DC:D0]	Reserved

Configuration space mapping rules:

- Configuration addresses (to “BusNo” and “Device” as specified by IOCF8 [IO-Space Configuration

[Address](#)] in the case of IO accesses or [2.7 \[Configuration Space\]](#) in the case of MMIO accesses) are within the defined range if:

( {BusNumBase[7:0]} <= BusNo <= {BusNumLimit[7:0]} ) & (DevCmpEn==0); or  
( {BusNumBase[4:0]} <= Device <= {BusNumLimit[4:0]} ) & (DevCmpEn==1) & (BusNo == 00h).

- Configuration regions must not overlap each other.
- The appropriate RE or WE bit(s) must be set.
- See [2.8.2.1.3 \[Configuration Space\]](#).

Bits	Description
31:24	<b>BusNumLimit[7:0]: bus number limit bits[7:0]</b> . Read-write. Reset: 0.
23:16	<b>BusNumBase[7:0]: bus number base bits[7:0]</b> . Read-write. Reset: 0.
15:3	Reserved.
2	<b>DevCmpEn: device number compare mode enable</b> . Read-write. Reset: 0. 1=A device number range rather than a bus number range is used to specify the configuration-space window (see above).
1	<b>WE: write enable</b> . Read-write. Reset: 0. 1=Writes to this configuration-space address range are enabled.
0	<b>RE: read enable</b> . Read-write. Reset: 0. 1=Reads to this configuration-space address range are enabled.

#### D18F1xF0 DRAM Hole Address

IF (D18F2x118[LockDramCfg]) THEN Read-only. ELSE Read-write. ENDIF. Same-for-all. Reset: 0000\_0000h. See [2.9.10 \[Memory Hoisting\]](#).

Bits	Description
31:24	<b>DramHoleBase[31:24]: DRAM hole base address</b> . Specifies the base address of the IO hole, below the 4GB address level, that is used in memory hoisting. Normally, DramHoleBase >= <a href="#">MSRC001_001A[TOM[31:24]]</a> .
23:16	Reserved.
15:7	<b>DramHoleOffset[31:23]: DRAM hole offset address</b> . When <a href="#">D18F1x2[1,0][8,0][LgcyMmioHoleEn]</a> ==1, this offset is subtracted from the physical address of certain accesses in forming the normalized address.
6:3	Reserved.
2	Reserved.
1	<b>DramMemHoistValid: dram memory hoist valid</b> . 1=Memory hoisting for the address range is enabled. 0=Memory hoisting is not enabled. This bit should be set if any <a href="#">D18F1x2[1,0][8,0][LgcyMmioHoleEn]</a> ==1 or DramHoleBase != 0.
0	<b>DramHoleValid: dram hole valid</b> . 1=Memory hoisting is enabled in the node. 0=Memory hoisting is not enabled. This bit should be set in the node that owns the DRAM address space that is hoisted above the 4 GB address level. See DramHoleBase.

#### D18F1xF4 VGA Enable

Reset: 0000\_0000h. All these bits are read-write unless Lock is set.

Bits	Description										
31:15	Reserved.										
14	<b>DstSubLink: destination sublink.</b> Read-write. When a link is ungang, this bit specifies the destination sublink of the link specified by <a href="#">D18F1xF4</a> [DstLink]. 0=The destination link is sublink 0. 1=The destination link is sublink 1. If the link is ganged, then this bit must be low.										
13:12	<b>DstLink: destination link ID.</b> Read-write. For transactions within the <a href="#">D18F1xF4</a> [VE]-defined ranges, this field specifies the destination IO link number of the destination node. <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Link 0</td> </tr> <tr> <td>01b</td> <td>Link 1</td> </tr> <tr> <td>10b</td> <td>Link 2</td> </tr> <tr> <td>11b</td> <td>Link 3</td> </tr> </tbody> </table>	Bits	Description	00b	Link 0	01b	Link 1	10b	Link 2	11b	Link 3
Bits	Description										
00b	Link 0										
01b	Link 1										
10b	Link 2										
11b	Link 3										
11:7	Reserved.										
6:4	<b>DstNode: destination node ID.</b> Read-write. For transactions within the <a href="#">D18F1xF4</a> [VE]-defined range, this field specifies the destination node ID.										
3	<b>Lock.</b> Read-write. 1=All the bits in this register ( <a href="#">D18F1xF4</a> ) are read-only (including this bit).										
2	Reserved.										
1	<b>NP: non-posted.</b> Read-write. 1=CPU write requests to the <a href="#">D18F1xF4</a> [VE]-defined MMIO range are passed through the non-posted channel. 0=CPU writes may be posted.										
0	<b>VE: VGA enable.</b> Read-write. 1=Transactions targeting the VGA-compatible address space are routed and controlled as specified by this register. 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 64 KB of IO space; i.e., address bits[24:16] must be low). 0=Transactions targeting the VGA-compatible address space are not affected by the state of this register. When this bit is set, the state of <a href="#">D18F1xF4</a> [VE] is ignored.										

### D18F1x10C DCT Configuration Select

Reset: 0000\_0000h.

Bits	Description										
31:6	Reserved.										
5:4	<b>NbPsSel: NB P-state configuration select.</b> Read-write. Specifies the set of DCT Pstate registers to which accesses are routed. <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>NB P-state 0</td> </tr> <tr> <td>01b</td> <td>NB P-state 1</td> </tr> <tr> <td>10b</td> <td>NB P-state 2</td> </tr> <tr> <td>11b</td> <td>NB P-state 3</td> </tr> </tbody> </table> <p>The following registers must be programmed for each NB P-state enabled by <a href="#">D18F5x16</a>[C:0][NbPstateEn]:</p> <ul style="list-style-type: none"> <li>• <a href="#">D18F2x210_dct</a>[0]_nbp[3:0][MaxRdLatency, DataTxFifoWrDly].</li> <li>• <a href="#">D18F2x210_dct</a>[0]_nbp[3:0][RdPtrInit].</li> </ul>	Bits	Description	00b	NB P-state 0	01b	NB P-state 1	10b	NB P-state 2	11b	NB P-state 3
Bits	Description										
00b	NB P-state 0										
01b	NB P-state 1										
10b	NB P-state 2										
11b	NB P-state 3										

3	<p><b>MemPsSel: Memory P-state configuration select.</b> Read-write. Specifies the set of DCT controller registers to which accesses are routed. This register works independently of NbPsSel. 0=Memory P-state 0. 1=Memory P-state 1. See 2.5.7.1 [Memory P-states] and 2.9.3 [DCT Configuration Registers]. The following registers must be programmed for each memory P-state enabled by <a href="#">D18F5x16[C:0][MemPstate]</a>:</p> <ul style="list-style-type: none"> <li>• <a href="#">D18F2x2E8_dct[0]_mp[1:0]</a></li> <li>• <a href="#">D18F2x2EC_dct[0]_mp[1:0]</a></li> </ul>						
2:0	<p><b>DctCfgSel: DRAM controller configuration select.</b> Read-write. Specifies DCT controller to which accesses are routed. See 2.9.3 [DCT Configuration Registers].</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">Bits</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>DCT 0</td> </tr> <tr> <td>111b-001b</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Description	000b	DCT 0	111b-001b	Reserved
Bits	Description						
000b	DCT 0						
111b-001b	Reserved						

**D18F1x120 DRAM Base System Address**

IF ([D18F2x118\[LockDramCfg\]](#)) THEN Read-only. ELSE Read-write. ENDIF. Cold reset: 0000\_0000h. [D18F1x120](#) and [D18F1x124](#) are required to specify the base and limit system address range of the DRAM connected to the local node.

DRAM accesses to the local node with physical address `Addr[47:0]` that are within the following range are directed to the DCTs:

$$\{\text{DramBaseAddr}[47:27], 000\_0000\text{h}\} \leq \text{Addr}[47:0] \leq \{\text{DramLimitAddr}[47:27], 7\text{FF\_FFF}\text{h}\};$$

DRAM accesses to the local node that are outside of this range are master aborted.

The address of the DRAM transaction is normalized before passing it to the DCTs by subtracting `DramBaseAddr`.

This range is also used to specify the range of DRAM covered by the scrubber (see [D18F3x58](#) and [D18F3x5C](#)).

Bits	Description
31:21	Reserved.
20:0	<b>DramBaseAddr[47:27]</b> .

**D18F1x124 DRAM Limit System Address**

IF ([D18F2x118\[LockDramCfg\]](#)) THEN Read-only. ELSE Read-write. ENDIF. See [D18F1x120 \[DRAM Base System Address\]](#).

Bits	Description
31:24	Reserved
23:21	Reserved.
20:0	<b>DramLimitAddr[47:27]</b> . Cold reset: 1F_FFFFh.

**D18F1x2[1C:00] DRAM Controller Base/Limit**

The DRAM controller base and limit registers define a DRAM controller address range and specify the mapping of physical DRAM addresses to a DCT as selected by DctSel or DctIntLvEn. The following base/limit register pairs specify the address ranges:

Table 120: Register Mapping for D18F1x2[1C:00]

Function	Base Address	Limit Address
Range 0	D18F1x200	D18F1x204
Reserved	D18F1x208, D18F1x21[8,0]	D18F1x20C, D18F1x21[C,4]

BIOS should observe the following DCT configuration requirements:

- DRAM addresses are within the defined range if:  
{DctBaseAddr[39:27], 000b, 00\_0000h} <= address[39:0] <= {DctLimitAddr[39:27], 111b, FF\_FFFFh}.
- DCT base/limit address ranges must not overlap each other.
- A maximum of two address ranges may be mapped to a single DCT.

**Hoisting.** When memory hoisting is enabled via LegacyMmioHoleEn, the corresponding DctBaseAddr/DctLimitAddr should be configured to account for the memory hoisted above the hole. A contiguous memory hole should only be mapped by one DctBaseAddr/DctLimitAddr pair. See 2.9.10 [Memory Hoisting].

**D18F1x2[1,0][8,0] DRAM Controller Base**

IF (D18F2x118[LockDramCfg]) THEN Read-only. ELSE Read-write. ENDIF. Reset: 0000\_0000h.

Table 121: Register Mapping for D18F1x2[1,0][8,0]

Register	Function
D18F1x200	Range 0
D18F1x208	Reserved
D18F1x21[8,0]	Reserved

Bits	Description						
31:24	Reserved.						
23:11	<b>DctBaseAddr[39:27]: DRAM controller base address [39:27].</b> Read-write. Specifies the base physical address bits for this address range.						
10:7	Reserved.						
6:4	<b>DctSel: DRAM controller select.</b> Read-write. Specifies the DCT mapped to this address range.  <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>DCT 0</td> </tr> <tr> <td>111b-010b</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Definition	000b	DCT 0	111b-010b	Reserved
Bits	Definition						
000b	DCT 0						
111b-010b	Reserved						
3	Reserved.						
2	Reserved.						

1	<b>LgcyMmioHoleEn: legacy mmio hole enable.</b> Read-write. BIOS: See <a href="#">2.9.10 [Memory Hoisting]</a> . 1=Enable memory hoisting for this address range. BIOS sets this bit for an address range that spans the 4GB boundary and contains a hole for addresses used by MMIO. 0=Memory hoisting is not enabled.
0	<b>DctAddrVal: DRAM controller address valid.</b> Read-write. 1=Specifies this address range is valid and enabled. 0=This address range is not enabled.

**D18F1x2[1,0][C,4] DRAM Controller Limit**

IF (D18F2x118[LockDramCfg]) THEN Read-only. ELSE Read-write. ENDIF. Reset: 0000\_0000h.

Table 122: [Register Mapping](#) for D18F1x2[1,0][C,4]

Register	Function
D18F1x204	Range 0
D18F1x20C	Reserved
D18F1x21[C,4]	Reserved

Bits	Description
31:24	Reserved.
23:11	<b>DctLimitAddr[39:27]: DRAM controller limit address bits [39:27].</b> Read-write. Specifies the limit physical address bits for this address range.
10:4	Reserved.
3:0	Reserved.

### 3.11 Device 18h Function 2 Configuration Registers

See 3.1 [Register Descriptions and Mnemonics]. See 2.7 [Configuration Space].

#### D18F2x00 Device/Vendor ID

Bits	Description
31:16	<b>DeviceID: device ID.</b> Read-only. Value: 1532h.
15:0	<b>VendorID: vendor ID.</b> Read-only. Value: 1022h.

#### D18F2x08 Class Code/Revision ID

Reset: 0600\_0000h.

Bits	Description
31:8	<b>ClassCode.</b> Read-only. Provides the host bridge class code as defined in the PCI specification.
7:0	<b>RevID: revision ID.</b> Read-only.

#### D18F2x0C Header Type

Reset: 0080\_0000h.

Bits	Description
31:0	<b>HeaderTypeReg.</b> Read-only. These bits are fixed at their default values. The header type field indicates that there multiple functions present in this device.

#### D18F2x[5C:40]\_dct[0] DRAM CS Base Address

IF (D18F2x118[LockDramCfg]) THEN Read-only. ELSE Read-write. ENDIF. Reset: 0000\_0000h. See 2.9.3 [DCT Configuration Registers].

These registers along with D18F2x[6C:60]\_dct[0] [DRAM CS Mask], translate DRAM request addresses (to a DRAM controller) into DRAM chip selects. Supported DIMM sizes are specified in D18F2x80\_dct[0] [DRAM Bank Address Mapping]. For more information on the DRAM controllers, see 2.9 [DRAM Controllers (DCTs)].

For each chip select, there is a DRAM CS Base Address register. For each CS pair there is a DRAM CS Mask Register. For each CS pair, an even CS must be populated if the odd CS is populated.

**Table 123: DIMM, Chip Select, and Register Mapping**

Base Address Registers	Mask Register	Logical DIMM	Chip Select <sup>1</sup>
D18F2x40_dct[x]	F2x60	0	MEMCS[x]_L[0]
D18F2x44_dct[x]			MEMCS[x]_L[1]

**Table 123: DIMM, Chip Select, and Register Mapping**

Base Address Registers	Mask Register	Logical DIMM	Chip Select <sup>1</sup>
D18F2x48_dct[x]	F2x64	1	MEMCS[x]_L[2]
D18F2x4C_dct[x]			MEMCS[x]_L[3]
D18F2x[5C:50]_dct[x]	F2x6[C,8]	Reserved	
1. See 2.9.4 [DDR Pad to Processor Pin Mapping]			

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 [Northbridge (NB)].

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 2.9.9.1 [Chip Select Interleaving]. The hardware supports the use of lower-order address bits to interleave chip selects if (1) the each chip select of the memory system spans the same amount of memory and (2) the number of chip selects of the memory system is a power of two.

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[38:27] & ~AddrMask[i][38:27]),
    (InputAddr[21:11] & ~AddrMask[i][21:11]) } ==
  { (BaseAddr[i][38:27] & ~AddrMask[i][38:27]),
    (BaseAddr[i][21:11] & ~AddrMask[i][21:11]) } );
```

Bits	Description
31	Reserved.
30:19	<b>BaseAddr[38:27]: normalized physical base address bits [38:27].</b>
18:16	Reserved.
15:5	<b>BaseAddr[21:11]: normalized physical base address bits [21:11].</b>
4	Reserved.



3	<b>OnDimmMirror: on-DIMM mirroring (ODM) enabled.</b> 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. Hardware bit swapping does not occur for commands sent via <code>D18F2x7C_dct[0][SendMrsCmd]</code> when <code>D18F2x7C_dct[0][EnDramInit] = 0</code> . 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. The following bits are swapped when enabled: <ul style="list-style-type: none"> <li>• BA0 and BA1.</li> <li>• A3 and A4.</li> <li>• A5 and A6.</li> <li>• A7 and A8.</li> </ul>
2	<b>TestFail: memory test failed.</b> 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 <code>CSEnable=1</code> and <code>TestFail=1</code> as mutually exclusive.
1	Reserved.
0	<b>CSEnable: chip select enable.</b>

#### **D18F2x[6C:60]\_dct[0] DRAM CS Mask**

IF (`D18F2x118[LockDramCfg]`) THEN Read-only. ELSE Read-write. ENDIF. Reset: 0000\_0000h. See 2.9.3 [DCT Configuration Registers]. See `D18F2x[5C:40]_dct[0]`.

Bits	Description
31	Reserved.
30:19	<b>AddrMask[38:27]: normalized physical address mask bits [38:27].</b>
18:16	Reserved.
15:5	<b>AddrMask[21:11]: normalized physical address mask bits [21:11].</b>
4	Reserved.
3:2	Reserved.
1:0	Reserved.

#### **D18F2x78\_dct[0] DRAM Control**

See 2.9.3 [DCT Configuration Registers].

Bits	Description
31	Reserved.
30:18	Reserved.
17	<b>AddrCmdTriEn: address command tristate enable.</b> Read-write. Reset: 0. BIOS: See 2.9.7.7. 1=Tristate the address, command, and bank buses when a Deselect command is issued.
16	Reserved.

15	Reserved.
14:11	Reserved.
10:0	Reserved.

### **D18F2x7C\_dct[0] DRAM Initialization**

Reset: 0000\_0000h. See 2.9.3 [DCT Configuration Registers]. See 2.9.7.8 [DRAM Device and Controller Initialization].

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. BIOS should not assert LDTSTOP\_L while EnDramInit is set. Setting more than one of the command bits in this register (SendControlWord, SendMrsCmd, and SendAutoRefresh) at a time results in undefined behavior.

Bits	Description
31	<b>EnDramInit: enable DRAM initialization.</b> Read-write. 1=Place the DRAM controller in the BIOS-controlled DRAM initialization mode. The DCT deasserts CKE when this bit is set. BIOS must wait until <a href="#">D18F2x98_dct[0][DctAccessDone]</a> = 1 before programming AssertCke=1 and DeassertMemRstX=1. BIOS must clear this bit after DRAM initialization is complete. BIOS must not set this bit on a DCT with no attached DIMMs.
30	Reserved.
29	<b>SendZQCmd: send ZQ command.</b> Read; write-1-only; cleared-by-hardware. 1=The DCT sends the ZQ calibration command with either all even or all odd chip selects active. The first command targets even chip selects. Subsequent commands alternate between even and odd chip selects. This bit is cleared by the hardware after the command completes. This bit is valid only when EnDramInit=1.
28	<b>AssertCke: assert CKE.</b> Read-write. Setting this bit causes the DCT to assert the CKE pins. This bit cannot be used to deassert the CKE pins.
27	<b>DeassertMemRstX: deassert memory reset.</b> 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	<b>SendMrsCmd: send MRS command.</b> Read; write-1-only; cleared-by-hardware. 1=The DCT sends the MRS commands defined by the MrsChipSel, MrsAddress, and MrsBank fields of this register. This bit is cleared by hardware after the command completes. Reserved if <a href="#">D18F2x78_dct[0][AddrCmdTriEn]</a> =1.
25	<b>SendAutoRefresh: send auto refresh command.</b> Read; write-1-only; cleared-by-hardware. 1=The DCT sends an auto refresh command. This bit is cleared by hardware after the command completes.
24	Reserved.

23:21	<b>MrsChipSel: MRS command chip select.</b> Read-write. Specifies which DRAM chip select is used for MRS commands. Defined only if (~EnDramInit   ~D18F2x90_dct[0][UnbuffDimm]); otherwise MRS commands are sent to all chip selects. <u>Bits</u> 000b 110b-001b 111b	<u>Description</u> MRS command is sent to CS0 MRS command is sent to CS<MrsChipSel> MRS command is sent to CS7
20:18	<b>MrsBank[2:0]: bank address for MRS commands.</b> Read-write. Specifies the data driven on the DRAM bank pins for MRS commands.	
17:0	<b>MrsAddress[17:0]: address for MRS commands.</b> Read-write. Specifies the data driven on the DRAM address pins for MRS commands.	

**D18F2x80\_dct[0] DRAM Bank Address Mapping**

IF (D18F2x118[LockDramCfg]) THEN Read-only. ELSE Read-write. ENDIF. Reset: 0000\_0000h. See 2.9.3 [DCT Configuration Registers]. These fields specify DIMM configuration information. These fields are required to be programmed based on the DRAM device size and with information of the DIMM. Table 124 shows the bit numbers for each position.

Bits	Description
31:16	Reserved.
15:8	Reserved.
7:4	<b>DimmAddrMap1: DIMM 1 address map.</b>
3:0	<b>DimmAddrMap0: DIMM 0 address map.</b>

**Table 124: 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 1Gb, x16	15	14	13	Row	x	x	x	17	16	28	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	
0011b		Reserved				Row																	
						Col																	
0100b		Reserved				Row																	
						Col																	
0101b	1GB	1Gb, x8 2Gb, x16	15	14	13	Row	x	x	17	16	29	28	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	
0110b		Reserved				Row																	
						Col																	
0111b	2GB	2Gb, x8 4Gb, x16	15	14	13	Row	x	17	16	30	29	28	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	
1000b		Reserved				Row																	
						Col																	

**Table 124: 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		
1001b		Reserved																						
1010b	4GB	4Gb, x8 8Gb, x16	15	14	13	Row	17	16	31	30	29	28	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		
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		
1111b-1100b		Reserved																						

**D18F2x84\_dct[0] DRAM MRS**

Reset: 0000\_0005h. See 2.9.3 [DCT Configuration Registers].

Bits	Description																												
31:24	Reserved.																												
23	<p><b>PchgPDMoSel: precharge power down mode select.</b> Read-write. BIOS: 1. Specifies how a chip select enters and exits power down mode. This mode is enabled by <a href="#">D18F2x94_dct[0][PowerDownEn]</a> and its behavior varies based on the setting of <a href="#">D18F2x94_dct[0][PowerDownMode]</a> and <a href="#">MR0[PPD]</a> in <a href="#">D18F2x2E8_dct[0]_mp[1:0][MxMr0]</a>.</p> <table border="1"> <thead> <tr> <th><u>PowerDownMode</u></th> <th><u>PchgPDMoSel</u></th> <th><u>MR0[PPD]</u></th> <th><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>0b</td> <td>0b</td> <td>0b</td> <td>Full channel slow exit (DLL off)</td> </tr> <tr> <td>0b</td> <td>0b</td> <td>1b</td> <td>Full channel fast exit (DLL on)</td> </tr> <tr> <td>0b</td> <td>1b</td> <td>xb</td> <td>Full channel dynamic fast exit/slow exit</td> </tr> <tr> <td>1b</td> <td>0b</td> <td>0b</td> <td>Reserved</td> </tr> <tr> <td>1b</td> <td>0b</td> <td>1b</td> <td>Partial channel fast exit (DLL on)</td> </tr> <tr> <td>1b</td> <td>1b</td> <td>xb</td> <td>Partial channel dynamic fast exit/slow exit</td> </tr> </tbody> </table> <p>See <a href="#">D18F2x248_dct[0]_mp[1:0][Txpdll, Txp]</a>. In dynamic fast exit/slow exit power down mode, the DCT dynamically issues MRS command(s) to the DRAM to specify the powerdown mode; the DCT specifies fast exit mode when chip selects on one of CKEs has recently been active; it specifies deep power down when chip selects on all CKEs have been idle. PchgPDMoSel=0 &amp;&amp; MR0[PPD]=1 fast exit modes are reserved if S3 is also supported .</p>	<u>PowerDownMode</u>	<u>PchgPDMoSel</u>	<u>MR0[PPD]</u>	<u>Description</u>	0b	0b	0b	Full channel slow exit (DLL off)	0b	0b	1b	Full channel fast exit (DLL on)	0b	1b	xb	Full channel dynamic fast exit/slow exit	1b	0b	0b	Reserved	1b	0b	1b	Partial channel fast exit (DLL on)	1b	1b	xb	Partial channel dynamic fast exit/slow exit
<u>PowerDownMode</u>	<u>PchgPDMoSel</u>	<u>MR0[PPD]</u>	<u>Description</u>																										
0b	0b	0b	Full channel slow exit (DLL off)																										
0b	0b	1b	Full channel fast exit (DLL on)																										
0b	1b	xb	Full channel dynamic fast exit/slow exit																										
1b	0b	0b	Reserved																										
1b	0b	1b	Partial channel fast exit (DLL on)																										
1b	1b	xb	Partial channel dynamic fast exit/slow exit																										
22:2	Reserved.																												
1:0	<p><b>BurstCtrl: burst length control.</b> Read-write. BIOS: 01b. Specifies the number of sequential beats of DQ related to one read or write command. Requests from the processor are always 64-byte-length. Requests generated by <a href="#">D18F2x250_dct[0]</a> are always 64-byte-length. Requests from GMC may be 32-byte or 64-byte-length. Software must ensure that GMC requests are disabled to configure the DCT and DRAMs for 8-beat burst length (e.g. during training). If this mode is changed, software must issue a mode-register set command to MR0 of the DRAMs to place them in the same mode.</p> <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>8 beats</td> </tr> <tr> <td>01b</td> <td>Dynamic 4 or 8 beats</td> </tr> <tr> <td>11b-10b</td> <td>Reserved</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Description</u>	00b	8 beats	01b	Dynamic 4 or 8 beats	11b-10b	Reserved																				
<u>Bits</u>	<u>Description</u>																												
00b	8 beats																												
01b	Dynamic 4 or 8 beats																												
11b-10b	Reserved																												

**D18F2x88\_dct[0] DRAM Timing Low**

Reset: 3F00\_0000h. See 2.9.3 [DCT Configuration Registers].

Bits	Description														
31:30	Reserved.														
29:24	<b>MemClkDis: MEMCLK disable.</b> Read-write. 1=Disable the MEMCLK. 0=Enable MEMCLK. All enabled clocks should be 0; all no-connect and unused clocks should be 1. <table border="1"> <thead> <tr> <th>Bit</th> <th>Pad</th> </tr> </thead> <tbody> <tr> <td>[0]</td> <td>MEMCLK_H[0]</td> </tr> <tr> <td>[1]</td> <td>MEMCLK_H[1]</td> </tr> <tr> <td>[2]</td> <td>MEMCLK_H[2]</td> </tr> <tr> <td>[3]</td> <td>MEMCLK_H[3]</td> </tr> <tr> <td>[4]</td> <td>MEMCLK_H[4]</td> </tr> <tr> <td>[5]</td> <td>MEMCLK_H[5]</td> </tr> </tbody> </table>	Bit	Pad	[0]	MEMCLK_H[0]	[1]	MEMCLK_H[1]	[2]	MEMCLK_H[2]	[3]	MEMCLK_H[3]	[4]	MEMCLK_H[4]	[5]	MEMCLK_H[5]
Bit	Pad														
[0]	MEMCLK_H[0]														
[1]	MEMCLK_H[1]														
[2]	MEMCLK_H[2]														
[3]	MEMCLK_H[3]														
[4]	MEMCLK_H[4]														
[5]	MEMCLK_H[5]														
23:0	Reserved.														

**D18F2x8C\_dct[0] DRAM Timing High**

Reset: 0000\_0000h. See 2.9.3 [DCT Configuration Registers].

Bits	Description										
31:19	Reserved.										
18	<b>DisAutoRefresh: disable automatic refresh.</b> Read-write. BIOS: See 2.9.7.7. 1=Automatic refresh is disabled.										
17:16	<b>Tref: refresh rate.</b> Read-write. BIOS: See 2.9.7.5. This specifies the average time between refresh requests to all DRAM devices. <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <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 us</td> </tr> <tr> <td>11b</td> <td>Every 3.9 us</td> </tr> </tbody> </table>	Bits	Description	00b	Undefined behavior.	01b	Reserved	10b	Every 7.8 us	11b	Every 3.9 us
Bits	Description										
00b	Undefined behavior.										
01b	Reserved										
10b	Every 7.8 us										
11b	Every 3.9 us										
15:0	Reserved.										

**D18F2x90\_dct[0] DRAM Configuration Low**

See 2.9.3 [DCT Configuration Registers].

Bits	Description						
31:28	<b>IdleCycLimit: idle cycle limit.</b> Read-write. Reset: 8h. BIOS: 8h. Specifies the number of MEMCLK cycles an idle page is open before it is closed if DynPageCloseEn=0. This field is ignored if DynPageCloseEn=1. <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>8 clocks</td> </tr> <tr> <td>Fh-1h</td> <td>&lt;IdleCycLimit&gt;*16 clocks</td> </tr> </tbody> </table>	Bits	Description	0h	8 clocks	Fh-1h	<IdleCycLimit>*16 clocks
Bits	Description						
0h	8 clocks						
Fh-1h	<IdleCycLimit>*16 clocks						

27	<b>DisDllShutdownSR: disable DLL shutdown in self-refresh mode.</b> Read-write. Reset: 1. BIOS: See <a href="#">2.9.7.7</a> . 1=Disable the power saving features of shutting down DDR phy DLLs during DRAM self refresh and memory P-states. 0=Shutdown DLLs during DRAM self refresh and allow memory P-state transitions.										
26	Reserved.										
25	<b>PendRefPaybackS3En: pending refresh payback S3 enable.</b> Read-write. Reset: 0. BIOS: 1. Specifies the S3 refresh payback behavior when PendRefPayback=0. 1=Pending refreshes are paid back on S3 entry. 0=Pending refreshes are not paid back on S3 entry.										
24	<b>StagRefEn: Stagger Refresh Enable.</b> Read-write. Reset: 0. BIOS: 1.1=The DRAM controller arbitrates refreshes among chip selects based on the Tstag value . See <a href="#">D18F2x228_dct[0]</a> . 0=DCT arbitrates among chip selects using the Trfc value. See <a href="#">D18F2x208_dct[0]</a> .										
23	<b>ForceAutoPchg: force auto precharging.</b> Read-write. Reset: 0. BIOS: See <a href="#">2.9.7.7</a> . 1=Force auto-precharge cycles with every read or write command.										
22:21	<b>IdleCycLowLimit: idle cycle low limit.</b> Read-write. Reset: 0. Specifies the number of MEMCLK cycles a page is allowed to be open before it may be closed by the dynamic page close logic. This field is ignored if <a href="#">D18F2x90_dct[0][DynPageCloseEn]</a> = 0. <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>16 clocks</td> </tr> <tr> <td>01b</td> <td>32 clocks</td> </tr> <tr> <td>10b</td> <td>64 clocks</td> </tr> <tr> <td>11b</td> <td>96 clocks</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Description</u>	00b	16 clocks	01b	32 clocks	10b	64 clocks	11b	96 clocks
<u>Bits</u>	<u>Description</u>										
00b	16 clocks										
01b	32 clocks										
10b	64 clocks										
11b	96 clocks										
20	<b>DynPageCloseEn: dynamic page close enable.</b> Read-write. Reset: 0. See <a href="#">2.9.7.7 [DCT Training Specific Configuration]</a> . 1=The DRAM controller dynamically determines when to close open pages based on the history of that particular page and <a href="#">D18F2x90_dct[0][IdleCycLowLimit]</a> . 0=Any open pages not auto-pre-charged by the DRAM controller are automatically closed after IdleCycLimit clocks of inactivity.										
19	<b>DimmEccEn: DIMM ECC enable.</b> Read-write. Reset: 0.1=ECC checking is capable of being enabled for all DIMMs on the DRAM controller by <a href="#">D18F3x44[DramEccEn]</a> . This bit should not be set unless all populated DIMMs support ECC check bits. 0=ECC checking is disabled on the DRAM controller.										
18	<b>PendRefPayback: pending refresh payback.</b> Read-write. Reset: 0. BIOS: 0. 1=The DRAM controller executes all pending refresh commands before entering the self refresh state. 0=The controller enters the self refresh state regardless of the number of pending refreshes; applies to any self refresh entry if PendRefPaybackS3En=0, else any non-S3 self refresh entry.										
17	<b>EnterSelfRef: enter self refresh command.</b> Read, write-1-only; cleared-by-hardware. Reset: 0. 1=The DRAM controller places the DRAMs into self refresh mode. The DRAM interface is tristated 1 MEMCLK after the self refresh command is issued to the DRAMs. Once entered, the DRAM interface must remain in self refresh mode for a minimum of 5 MEMCLKs. This bit is read as a 1 while the enter-self-refresh command is executing; it is read as 0 at all other times.										
16	<b>UnbuffDimm: unbuffered DIMM.</b> Read-write or read-only, depending on the product. Reset: Product-specific. BIOS: 1. 1=The DRAM controller is connected to unbuffered DIMMs. 0=Reserved.										

15:12	Reserved.
11	Reserved.
10	Reserved.
9	Reserved.
8	Reserved.
7	Reserved.
6:2	Reserved.
1	<b>ExitSelfRef: exit self refresh (after suspend to RAM or for DRAM training) command.</b> Read, write-1-only; cleared-by-hardware. Reset: 0. Writing a 1 to this bit causes the DRAM controller to bring the DRAMs out of self refresh mode. It also causes the DRAM controller to issue ZQCL and MRS MR0 commands. This command should be executed by BIOS when returning from the suspend to RAM state, after the DRAM controller configuration registers are properly initialized, or when self refresh is used during DRAM training. This bit is read as a 1 while the exit-self-refresh command is executing; it is read as 0 at all other times. This bit should not be set if the DCT is disabled.
0	Reserved.

### D18F2x94\_dct[0] DRAM Configuration High

See 2.9.3 [DCT Configuration Registers].

Bits	Description						
31	<b>DphyMemPsSelEn: Ddr phy MemPsSel enable.</b> Read-write. Reset: 1h. BIOS: 1. 1=The DCT uses <a href="#">D18F1x10C[MemPsSel]</a> to select the memory P-state context of a phy register accessed by software with <code>DctOffset[29:20]=0D0h</code> . <code>DctOffset[6]</code> is ignored. 0=Software accesses use <code>DctOffset[6]</code> to select context of those registers. If <code>DctOffset[29:20]!=0D0h</code> then this register has no effect. See <a href="#">D18F2x9C_x0D04_E008_dct[0][PStateToAccess]</a> .						
30:29	Reserved.						
28:24	<b>DcqBypassMax: DRAM controller queue bypass maximum.</b> Read-write. Reset: 0h. BIOS: <a href="#">2.9.7.7</a> . 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. <table border="0"> <tr> <td><u>Bits</u></td> <td><u>Description</u></td> </tr> <tr> <td>0h</td> <td>No bypass; the oldest request is never bypassed.</td> </tr> <tr> <td>1Fh-1h</td> <td>The oldest request may be bypassed no more than &lt;DcqBypassMax&gt; time.</td> </tr> </table>	<u>Bits</u>	<u>Description</u>	0h	No bypass; the oldest request is never bypassed.	1Fh-1h	The oldest request may be bypassed no more than <DcqBypassMax> time.
<u>Bits</u>	<u>Description</u>						
0h	No bypass; the oldest request is never bypassed.						
1Fh-1h	The oldest request may be bypassed no more than <DcqBypassMax> time.						
23	<b>ProcOdtDis: processor on-die termination disable.</b> Read-write. Reset: 0h. 1=The processor-side on-die termination is disabled. 0=Processor-side on-die termination enabled. Changes to this bit must be performed prior to setting <code>MemClkFreqVal</code> .						

22	<p><b>BankSwizzleMode: bank swizzle mode.</b> Read-write. Reset: 0. BIOS: 2.9.7.7. 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 <a href="#">D18F2x80_dct[0]</a>, are remapped as follows:</p> <ul style="list-style-type: none"> <li>• Define X as a bank address bit (e.g., X=15 if the bank bit is specified to be address bit 15).</li> <li>• Define S(n) as the state of address bit n (0 or 1) and B as the remapped bank address bit. Then, <math>B = S(X) \wedge S(X + 3) \wedge S(X + 6)</math>; for an 8-bank DRAM.</li> </ul> <p>For example, encoding 02h of <a href="#">Table 124</a> would be remapped from Bank[2:0]={A15, A14, A13} to the following: Bank[2:0] = {A15^A18^A21, A14^A17^A20, A13^A16^A19}.</p>
21	<p><b>FreqChgInProg: frequency change in progress.</b> Read-only. Reset: 0. 1=A MEMCLK frequency change is in progress. The DDR phy asserts this bit when it is in the process of locking the PLL. BIOS should not program the phy registers while this bit is set. 0=DRAM-interface commands can be sent to the phy.</p>
20	<p><b>SlowAccessMode: slow access mode (a.k.a. 2T mode).</b> Read-write. Reset: 0. 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. If memory P-states are enabled then BIOS must set this bit if 2T timing is recommended for either memory P-state.</p>
19	Reserved. Reset: 1.
18	Reserved.
17	Reserved.
16	<p><b>PowerDownMode: power down mode.</b> Read-write. Reset: 0. BIOS: 1. Specifies how a chip select or group of chip selects enters power down mode when enabled by <a href="#">D18F2x94_dct[0][PowerDownEn]</a>. A chip select enters power down mode when the DCT deasserts the CKE pin. The command and address signals tristate one MEMCLK after CKE deasserts. The DCT behavior varies based on the setting of <a href="#">D18F2x84_dct[0][PchgPDMModeSel]</a>. See also <a href="#">Table 123 [DIMM, Chip Select, and Register Mapping]</a>.</p> <p>0=Channel CKE control mode; the DRAM channel is placed in power down mode when all chip selects associated with the channel are idle; CKE pins for the channel operate in lock step in terms of placing the channel in power down mode.</p> <p>1=Chip select CKE control mode; the chip select group controlled by a CKE pin is placed in power down mode when all chip selects associated with the pin are idle.</p>
15	<p><b>PowerDownEn: power down mode enable.</b> Read-write. Reset: 0. BIOS: 2.9.7.7.</p> <p>1=Power down mode is enabled. Only precharge power down mode is supported, not active power down mode. See PowerDownMode, <a href="#">D18F2x84_dct[0][PchgPDMModeSel]</a>, <a href="#">D18F2xA8_dct[0][PrtlChPDEnhEn, AggrPDEn, PDPhyPSDis]</a>, and <a href="#">D18F2x248_dct[0]_mp[1:0][PchgPDEnDelay]</a>.</p>
14	<p><b>DisDramInterface: disable the DRAM interface.</b> Read-write. Reset: 0. 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.</p>



13	Reserved.										
12	Reserved.										
11:10	<p><b>ZqcsInterval: ZQ calibration short interval.</b> Read-write. Reset: 00b. BIOS: See <a href="#">2.9.7.7</a>. This field specifies the programmable interval for the controller to send out the DRAM ZQ calibration short command.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>ZQ calibration short command is disabled</td> </tr> <tr> <td>01b</td> <td>64 ms</td> </tr> <tr> <td>10b</td> <td>128 ms</td> </tr> <tr> <td>11b</td> <td>256 ms</td> </tr> </tbody> </table>	Bits	Description	00b	ZQ calibration short command is disabled	01b	64 ms	10b	128 ms	11b	256 ms
Bits	Description										
00b	ZQ calibration short command is disabled										
01b	64 ms										
10b	128 ms										
11b	256 ms										
9:8	Reserved.										
7	<p><b>MemClkFreqVal: memory clock frequency valid.</b> Read-write. Reset: 0. System BIOS should set this bit after setting up <a href="#">D18F2x94_dct[0][MemClkFreq]</a> 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 <a href="#">2.9.7.4.2 [DRAM Channel Frequency Change]</a>.</p>										
6:5	Reserved.										
4:0	<p><b>MemClkFreq: memory clock frequency.</b> Read-write. Reset: 000b. Specifies the frequency and rate of the DRAM interface (MEMCLK). See: <a href="#">Table 125 [Valid Values for Memory Clock Frequency Value Definition]</a>. The rate is twice the frequency. See <a href="#">D18F5x84[DdrMaxRate]</a> and <a href="#">D18F5x84[DdrMaxRateEnf]</a>. See MemClkFreqVal.</p>										

**Table 125: Valid Values** for Memory Clock Frequency Value Definition

Bits	Description
03h-00h	Reserved
04h	333 MHz. (667 MT/s)
05h	Reserved
06h	400 MHz. (800 MT/s)
09h-07h	Reserved
0Ah	533 MHz. (1066 MT/s)
0Dh-0Bh	Reserved
0Eh	667 MHz. (1333 MT/s)
11h-0Fh	Reserved
12h	800 MHz. (1600 MT/s)
15h-13h	Reserved
16h	933 MHz. (1866 MT/s)
1Fh-17h	Reserved
19h	1050 MHz. (2100 MT/s)
1Ah	1066 MHz. (2133 MT/s)
1Fh	Reserved

**D18F2x98\_dct[0] DRAM Controller Additional Data Offset**

Reset: 8000\_0000h. See 2.9.3 [DCT Configuration Registers].

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:

1. Write the register number to `D18F2x98_dct[0][DctOffset]` with `D18F2x98_dct[0][DctAccessWrite]=0`.
2. Read the register contents from `D18F2x9C_dct[0]`.

Writes:

1. Write all 32 bits of register data to `D18F2x9C_dct[0]` (individual byte writes are not supported).
2. Write the register number to `D18F2x98_dct[0][DctOffset]` with `D18F2x98_dct[0][DctAccessWrite]=1`.
  - The data will be delivered to the phy similar to a posted memory-write, and the write will complete without any further action. However, to ensure that the contents of the array register write have been delivered to the phy, software issues a subsequent configuration register read or write to any register in the northbridge. For example, reading `D18F2x98_dct[0]` will accomplish this.

Registers for which there is one instance per memory P-state (listed with “\_mp[1:0]” appended to the register mnemonic) use `D18F1x10C[MemPsSel]`, `D18F2x94_dct[0][DphyMemPsSelEn]`, and `D18F2x9C_x0D04_E008_dct[0][PStateToAccess]` for software accesses. BIOS programs these fields appropriately to ensure consistency of registers with controller fields and DDR phy fields.

- `D18F2x9C_x0000_0[3:0]0[3:1]_dct[0]_mp[1:0]` refers to all instances of the `D18F2x9C_x0000_0[3:0]0[3:1]` register.
- `D18F2x9C_x0000_0[3:0]0[3:1]_dct[0]_mp[1]` refers to the register for memory P-state 1 of either or both DCTs.
- It is recommended that BIOS program context sensitive registers in batches (training/restoring all registers of a context before selecting a new context). BIOS should do the following prior to a new “batch”:
  - Program `D18F2x94_dct[0][DphyMemPsSelEn]=1`.
  - Program `D18F2x9C_x0D04_E008_dct[0][PStateToAccess]=target`.
  - Program `D18F1x10C[MemPsSel]=target`.

Writes to any register in this additional address space causes the FIFO pointers to be reset. Therefore, it is recommended that only BIOS write these registers.

Reads or writes to any register in this additional address space collide with system self-refresh requests. Once power management is enabled software should temporarily disable power management prior to accessing these registers.

Bits	Description
31	Reserved.
30	<b>DctAccessWrite: DRAM controller read/write select.</b> RAZ; write. 0=Specifies a read access. 1=Specifies a write access.
29:0	<b>DctOffset: DRAM controller offset.</b> Read-write.

**D18F2x9C\_dct[0] DRAM Controller Additional Data Port**

Reset: 0000\_0000h. See `D18F2x98_dct[0]` for register access information. See 2.9.3 [DCT Configuration Registers].

Bits	Description
31:0	<b>Data.</b> Read-write.

### **D18F2x9C\_x0000\_0000\_dct[0]\_mp[1:0] DRAM Output Driver Compensation Control**

See 2.9.7.6.6 [DRAM Address Timing and Output Driver Compensation Control].

Bits	Description												
31	Reserved.												
30:28	<p><b>ProcOdt: processor on-die termination.</b> Read-write. Cold reset: 011b. Specifies the resistance of the on-die termination resistors. This field is valid only when <code>D18F2x94_dct[0][ProcOdtDis]=0</code>.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>240 ohms +/- 20%</td> </tr> <tr> <td>001b</td> <td>120 ohms +/- 20%</td> </tr> <tr> <td>010b</td> <td>80 ohms +/- 20%</td> </tr> <tr> <td>011b</td> <td>60 ohms +/- 20%</td> </tr> <tr> <td>111b-100b</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Description	000b	240 ohms +/- 20%	001b	120 ohms +/- 20%	010b	80 ohms +/- 20%	011b	60 ohms +/- 20%	111b-100b	Reserved
Bits	Description												
000b	240 ohms +/- 20%												
001b	120 ohms +/- 20%												
010b	80 ohms +/- 20%												
011b	60 ohms +/- 20%												
111b-100b	Reserved												
27:23	Reserved.												
22:20	<p><b>DqsDrvStren: DQS drive strength.</b> Read-write. Cold reset: 011b. Specifies the drive strength of the DQS pins.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>0.75x</td> </tr> <tr> <td>001b</td> <td>1.0x</td> </tr> <tr> <td>010b</td> <td>1.25x</td> </tr> <tr> <td>011b</td> <td>1.5x</td> </tr> <tr> <td>111b-100b</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Description	000b	0.75x	001b	1.0x	010b	1.25x	011b	1.5x	111b-100b	Reserved
Bits	Description												
000b	0.75x												
001b	1.0x												
010b	1.25x												
011b	1.5x												
111b-100b	Reserved												
19	Reserved.												
18:16	<p><b>DataDrvStren: data drive strength.</b> Read-write. Cold reset: 011b. This field specifies the drive strength of the DRAM data pins.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>0.75x</td> </tr> <tr> <td>001b</td> <td>1.0x</td> </tr> <tr> <td>010b</td> <td>1.25x</td> </tr> <tr> <td>011b</td> <td>1.5x</td> </tr> <tr> <td>111b-100b</td> <td>Reserved</td> </tr> </tbody> </table> <p>This field is applied to DM signals as well.</p>	Bits	Description	000b	0.75x	001b	1.0x	010b	1.25x	011b	1.5x	111b-100b	Reserved
Bits	Description												
000b	0.75x												
001b	1.0x												
010b	1.25x												
011b	1.5x												
111b-100b	Reserved												
15	Reserved.												

14:12	<p><b>ClkDrvStren: MEMCLK drive strength.</b> Read-write. Cold reset: 011b. This field specifies the drive strength of the MEMCLK pins.</p> <table> <thead> <tr> <th><u>Bits</u></th> <th><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>1.0x</td> </tr> <tr> <td>001b</td> <td>1.25x</td> </tr> <tr> <td>010b</td> <td>1.5x</td> </tr> <tr> <td>011b</td> <td>2.0x</td> </tr> <tr> <td>100b</td> <td>Reserved</td> </tr> <tr> <td>101b</td> <td>0.75x</td> </tr> <tr> <td>111b-110b</td> <td>Reserved</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Description</u>	000b	1.0x	001b	1.25x	010b	1.5x	011b	2.0x	100b	Reserved	101b	0.75x	111b-110b	Reserved
<u>Bits</u>	<u>Description</u>																
000b	1.0x																
001b	1.25x																
010b	1.5x																
011b	2.0x																
100b	Reserved																
101b	0.75x																
111b-110b	Reserved																
11	Reserved.																
10:8	<p><b>AddrCmdDrvStren: address/command drive strength.</b> Read-write. Cold reset: 011b. This field specifies the drive strength of the address, RAS, CAS, WE, bank and parity pins.</p> <table> <thead> <tr> <th><u>Bits</u></th> <th><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>1.0x</td> </tr> <tr> <td>001b</td> <td>1.25x</td> </tr> <tr> <td>010b</td> <td>1.5x</td> </tr> <tr> <td>011b</td> <td>2.0x</td> </tr> <tr> <td>100b</td> <td>Reserved</td> </tr> <tr> <td>101b</td> <td>0.75x</td> </tr> <tr> <td>111b-110b</td> <td>Reserved</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Description</u>	000b	1.0x	001b	1.25x	010b	1.5x	011b	2.0x	100b	Reserved	101b	0.75x	111b-110b	Reserved
<u>Bits</u>	<u>Description</u>																
000b	1.0x																
001b	1.25x																
010b	1.5x																
011b	2.0x																
100b	Reserved																
101b	0.75x																
111b-110b	Reserved																
7	Reserved.																
6:4	<p><b>CsOdtDrvStren: CS/ODT drive strength.</b> Read-write. Cold reset: 011b. This field specifies the drive strength of the CS and ODT pins.</p> <table> <thead> <tr> <th><u>Bits</u></th> <th><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>1.0x</td> </tr> <tr> <td>001b</td> <td>1.25x</td> </tr> <tr> <td>010b</td> <td>1.5x</td> </tr> <tr> <td>011b</td> <td>2.0x</td> </tr> <tr> <td>100b</td> <td>Reserved</td> </tr> <tr> <td>101b</td> <td>0.75x</td> </tr> <tr> <td>111b-110b</td> <td>Reserved</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Description</u>	000b	1.0x	001b	1.25x	010b	1.5x	011b	2.0x	100b	Reserved	101b	0.75x	111b-110b	Reserved
<u>Bits</u>	<u>Description</u>																
000b	1.0x																
001b	1.25x																
010b	1.5x																
011b	2.0x																
100b	Reserved																
101b	0.75x																
111b-110b	Reserved																
3	Reserved.																
2:0	<p><b>CkeDrvStren: CKE drive strength.</b> Read-write. Cold reset: 011b. This field specifies the drive strength of the CKE pins.</p> <table> <thead> <tr> <th><u>Bits</u></th> <th><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>1.0x</td> </tr> <tr> <td>001b</td> <td>1.25x</td> </tr> <tr> <td>010b</td> <td>1.5x</td> </tr> <tr> <td>011b</td> <td>2.0x</td> </tr> <tr> <td>100b</td> <td>Reserved</td> </tr> <tr> <td>101b</td> <td>0.75x</td> </tr> <tr> <td>111b-110b</td> <td>Reserved</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Description</u>	000b	1.0x	001b	1.25x	010b	1.5x	011b	2.0x	100b	Reserved	101b	0.75x	111b-110b	Reserved
<u>Bits</u>	<u>Description</u>																
000b	1.0x																
001b	1.25x																
010b	1.5x																
011b	2.0x																
100b	Reserved																
101b	0.75x																
111b-110b	Reserved																

### **D18F2x9C\_x0000\_0[3:0]0[3:1]\_dct[0]\_mp[1:0] DRAM Write Data Timing**

BIOS: See 2.9.7.9.4 [DQS Position Training].

**Table 126:** Index Mapping for D18F2x9C\_x0000\_0[3:0]0[3:1]\_dct[0]\_mp[1:0]

D18F2x98_dct[0][31:0]	Function
0000_0001h	DIMM/CS 0 Bytes 3-0
0000_0002h	DIMM/CS 0 Bytes 7-4
0000_0003h	DIMM/CS 0 ECC
0000_0101h	DIMM/CS 1 Bytes 3-0
0000_0102h	DIMM/CS 1 Bytes 7-4
0000_0103h	DIMM/CS 1 ECC
0000_0201h	Reserved/CS 2 Bytes 3-0
0000_0202h	Reserved/CS 2 Bytes 7-4
0000_0203h	Reserved/CS 2 ECC
0000_0301h	Reserved/CS 3 Bytes 3-0
0000_0302h	Reserved/CS 3 Bytes 7-4
0000_0303h	Reserved/CS 3 ECC

**Table 127:** Byte Lane Mapping for D18F2x9C\_x0000\_0[3:0]0[3:1]\_dct[0]\_mp[1:0]

Register	Bits			
	31:24	23:16	15:8	7:0
D18F2x9C_x0000_0[3:0]01_dct[0]_mp[1:0]	Byte3	Byte2	Byte1	Byte0
D18F2x9C_x0000_0[3:0]02_dct[0]_mp[1:0]	Byte7	Byte6	Byte5	Byte4
D18F2x9C_x0000_0[3:0]03_dct[0]_mp[1:0]	Reserved	Reserved	Reserved	ECC

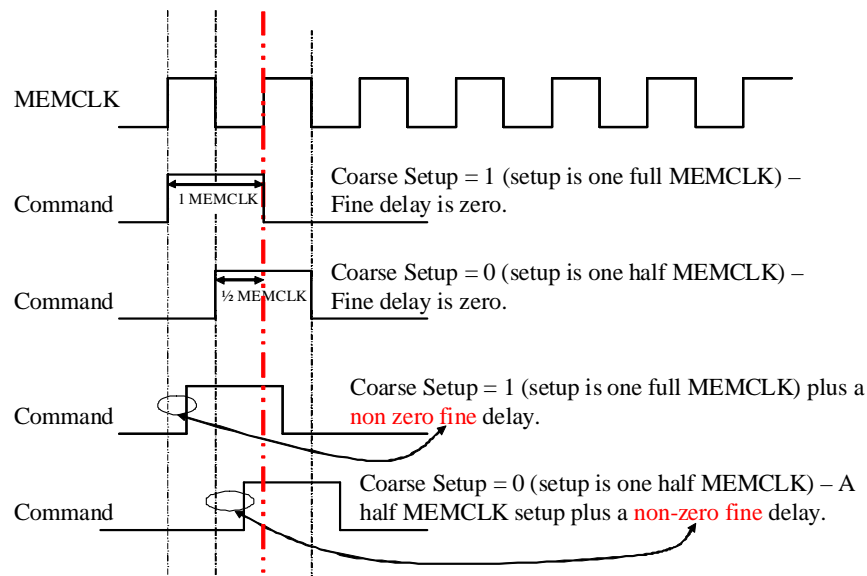
If D18F2xA8\_dct[0][PerRankTimingEn]=1 then the function is CS. Otherwise the function is DIMM. These registers control the timing of write DQ with respect to MEMCLK and allow transmit DQS to be centered in the data eye. The delay starts 1 UI before the rising edge of MEMCLK corresponding to the CAS-write-latency. See 2.9.7.9 [DRAM Training]. WrDatGrossDly must be programmed for a given DIMM and lane such that WrDatDly - WrDqsDly <= 0.5 MEMCLKs.

Bits	Description
31:29	<b>WrDatGrossDly: write data gross delay.</b> See: <a href="#">D18F2x9C_x0000_0[3:0]0[3:1]_dct[0]_mp[1:0][7:5]</a> .
28:24	<b>WrDatFineDly: write data fine delay.</b> See: <a href="#">D18F2x9C_x0000_0[3:0]0[3:1]_dct[0]_mp[1:0][4:0]</a> .
23:21	<b>WrDatGrossDly: write data gross delay.</b> See: <a href="#">D18F2x9C_x0000_0[3:0]0[3:1]_dct[0]_mp[1:0][7:5]</a> .
20:16	<b>WrDatFineDly: write data fine delay.</b> See: <a href="#">D18F2x9C_x0000_0[3:0]0[3:1]_dct[0]_mp[1:0][4:0]</a> .
15:13	<b>WrDatGrossDly: write data gross delay.</b> See: <a href="#">D18F2x9C_x0000_0[3:0]0[3:1]_dct[0]_mp[1:0][7:5]</a> .
12:8	<b>WrDatFineDly: write data fine delay.</b> See: <a href="#">D18F2x9C_x0000_0[3:0]0[3:1]_dct[0]_mp[1:0][4:0]</a> .

7:5	<b>WrDatGrossDly: write data gross delay.</b> Read-write. Reset: 0.
<u>Bits</u>	<u>Description</u>
000b	No delay
001b	0.5 MEMCLK delay
110b-010b	<WrDatGrossDly>/2 MEMCLK delay
111b	3.5 MEMCLK delay
4:0	<b>WrDatFineDly: write data fine delay.</b> Read-write. Cold reset: 0.
<u>Bits</u>	<u>Description</u>
00h	0/64 MEMCLK delay
1Eh-01h	<WrDatFineDly>/64 MEMCLK delay
1Fh	31/64 MEMCLK delay

**D18F2x9C\_x0000\_0004\_dct[0]\_mp[1:0] DRAM Address/Command Timing Control**

BIOS: 2.9.7.6.6.



**Figure 9: Address/Command Timing at the Processor Pins**

This register controls the timing of the address, command, chip select, ODT and clock enable pins with respect to MEMCLK as shown in Figure 9. See 2.9.7.4.2 [DRAM Channel Frequency Change] and 2.9.7.4.3 [Phy Fence Programming]. 2T timing is controlled by D18F2x94\_dct[0][SlowAccessMode]. If a setup time (coarse delay) field is changed and D18F2x94\_dct[0][MemClkFreqVal]=1, then software must toggle MemClkFreqVal for the delay to take effect.

Bits	Description
31:22	Reserved.
21	<b>AddrCmdSetup: address/command setup time.</b> Read-write. Reset: 0. 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=1 MEMCLK (2 MEMCLKs for 2T timing).

20:16	<b>AddrCmdFineDelay: address/command fine delay.</b> Specifies the time that the address and command pins are delayed from the default setup time. See: CkeFineDelay.								
15:14	Reserved.								
13	<b>CsOdtSetup: CS/ODT setup time.</b> Selects the default setup time for the CS and ODT pins versus MEMCLK. See: CkeSetup.								
12:8	<b>CsOdtFineDelay: CS/ODT fine delay.</b> Specifies the time that the CS and ODT pins are delayed from the default setup time. See: CkeFineDelay.								
7:6	Reserved.								
5	<b>CkeSetup: CKE setup time.</b> Read-write. Reset: 0. Selects the default setup time for the CKE pins versus MEMCLK. 0=1/2 MEMCLK. 1=1 MEMCLK.								
4:0	<p><b>CkeFineDelay: CKE fine delay.</b> Read-write. Cold reset: 00h. Specifies the time that the CKE pins are delayed from the default setup time.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00h</td> <td>0/64 MEMCLK delay</td> </tr> <tr> <td>1Eh-01h</td> <td>&lt;CkeFineDelay&gt;/64 MEMCLK delay</td> </tr> <tr> <td>1Fh</td> <td>31/64 MEMCLK delay</td> </tr> </tbody> </table>	Bits	Description	00h	0/64 MEMCLK delay	1Eh-01h	<CkeFineDelay>/64 MEMCLK delay	1Fh	31/64 MEMCLK delay
Bits	Description								
00h	0/64 MEMCLK delay								
1Eh-01h	<CkeFineDelay>/64 MEMCLK delay								
1Fh	31/64 MEMCLK delay								

**D18F2x9C\_x0000\_0[3:0]0[7:5]\_dct[0]\_mp[1:0] DRAM Read DQS Timing**

**Table 128: Index Mapping for D18F2x9C\_x0000\_0[3:0]0[7:5]\_dct[0]\_mp[1:0]**

D18F2x98_dct[0][31:0]	Function
0000_0005h	DIMM/CS 0 Bytes 3-0
0000_0006h	DIMM/CS 0 Bytes 7-4
0000_0007h	DIMM/CS 0 ECC
0000_0105h	DIMM/CS 1 Bytes 3-0
0000_0106h	DIMM/CS 1 Bytes 7-4
0000_0107h	DIMM/CS 1 ECC
0000_0205h	Reserved/CS 2 Bytes 3-0
0000_0206h	Reserved/CS 2 Bytes 7-4
0000_0207h	Reserved/CS 2 ECC
0000_0305h	Reserved/CS 3 Bytes 3-0
0000_0306h	Reserved/CS 3 Bytes 7-4
0000_0307h	Reserved/CS 3 ECC

**Table 129: Byte Lane Mapping for D18F2x9C\_x0000\_0[3:0]0[7:5]\_dct[0]\_mp[1:0]**

Register	Bits			
	29:25	21:17	13:9	5:1
D18F2x9C_x0000_0[3:0]05_dct[0]_mp[1:0]	Byte3	Byte2	Byte1	Byte0
D18F2x9C_x0000_0[3:0]06_dct[0]_mp[1:0]	Byte7	Byte6	Byte5	Byte4
D18F2x9C_x0000_0[3:0]07_dct[0]_mp[1:0]	Reserved	Reserved	Reserved	ECC

If D18F2xA8\_dct[0][PerRankTimingEn]=1 then the function is CS. Otherwise the function is DIMM. These registers control the timing of read (input) DQS signals with respect to DQ. See 2.9.7.9 [DRAM Training]. Writes to D18F2x9C\_x0000\_0[3:0]0[7:5]\_dct[0]\_mp[1:0] set

D18F2x9C\_x0D0F\_0[F,8:0]2[3:0]\_dct[0]\_mp[1:0][RdDqsTimeU, RdDqsTimeL] = {RdDqsTime, RdDqsTime}. Reads from D18F2x9C\_x0000\_0[3:0]0[7:5]\_dct[0]\_mp[1:0] return

D18F2x9C\_x0D0F\_0[F,8:0]2[3:0]\_dct[0]\_mp[1:0][RdDqsTimeL].

The actual delay applied to the DQS input signal before sampling data includes an internal part dependent delay plus the nominal delay specified by the register setting.

Bits	Description								
31:30	Reserved.								
29:25	<b>RdDqsTime: read DQS timing control.</b> See: <a href="#">D18F2x9C_x0000_0[3:0]0[7:5]_dct[0]_mp[1:0][5:1]</a> .								
24:22	Reserved.								
21:17	<b>RdDqsTime: read DQS timing control.</b> See: <a href="#">D18F2x9C_x0000_0[3:0]0[7:5]_dct[0]_mp[1:0][5:1]</a> .								
16:14	Reserved.								
13:9	<b>RdDqsTime: read DQS timing control.</b> See: <a href="#">D18F2x9C_x0000_0[3:0]0[7:5]_dct[0]_mp[1:0][5:1]</a> .								
8:6	Reserved.								
5:1	<b>RdDqsTime: read DQS timing control.</b> Read-write. Cold reset: 0Fh. <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00h</td> <td>0/64 MEMCLK delay</td> </tr> <tr> <td>1Eh-01h</td> <td>&lt;RdDqsTime&gt;/64 MEMCLK delay</td> </tr> <tr> <td>1Fh</td> <td>31/64 MEMCLK delay</td> </tr> </tbody> </table>	Bits	Description	00h	0/64 MEMCLK delay	1Eh-01h	<RdDqsTime>/64 MEMCLK delay	1Fh	31/64 MEMCLK delay
Bits	Description								
00h	0/64 MEMCLK delay								
1Eh-01h	<RdDqsTime>/64 MEMCLK delay								
1Fh	31/64 MEMCLK delay								
0	Reserved.								

### D18F2x9C\_x0000\_0008\_dct[0]\_mp[1:0] DRAM Phy Control

Cold reset: 0208\_0000h. See [2.9.7.9 \[DRAM Training\]](#).

Bits	Description										
31	Reserved.										
30	<b>DisAutoComp: disable automatic compensation.</b> Read-write. BIOS: See <a href="#">2.9.7.4.4</a> . 1=Disable the compensation control state machine. 0=The phy automatic compensation engine is enabled.										
29	<b>DisablePredriverCal: disable predriver calibration.</b> Read-write. BIOS: See <a href="#">2.9.7.4.4</a> . 1=Disable the update of predriver codes to all pads.										
28:14	Reserved.										
13	<b>DqsRcvTrEn: DQS receiver training enable.</b> Read-write. 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	<b>WrLvOdtEn: write levelization ODT enabled.</b> Read-write. 1=ODT specified by WrLvOdt is enabled during write levelization training. 0=ODT is disabled during write levelization training.										
11:8	<b>WrLvOdt: write levelization ODT.</b> Read-write. Specifies the state of the ODT pins when WrLvOdtEn is set. 1=ODT is enabled. 0=ODT is disabled. See <a href="#">2.9.7.6.5 [DRAM ODT Control]</a> . Tri-state enable for ODT is turned off by the phy when WrLvOdtEn=1. <table border="1"> <thead> <tr> <th>Bit</th> <th>Pad</th> </tr> </thead> <tbody> <tr> <td>[0]</td> <td>MEMODT[0][0]</td> </tr> <tr> <td>[1]</td> <td>MEMODT[0][1]</td> </tr> <tr> <td>[2]</td> <td>MEMODT[0][2]</td> </tr> <tr> <td>[3]</td> <td>MEMODT[0][3]</td> </tr> </tbody> </table>	Bit	Pad	[0]	MEMODT[0][0]	[1]	MEMODT[0][1]	[2]	MEMODT[0][2]	[3]	MEMODT[0][3]
Bit	Pad										
[0]	MEMODT[0][0]										
[1]	MEMODT[0][1]										
[2]	MEMODT[0][2]										
[3]	MEMODT[0][3]										



7:6	<b>FenceTrSel: fence train select.</b> Read-write. Specifies the flop to be used for phy based fence training. See PhyFenceTrEn. This field is shared by D18F2x9C_x0000_0008_dct[0]_mp0 and D18F2x9C_x0000_0008_dct[0]_mp1. <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>PRE flop</td> </tr> <tr> <td>01b</td> <td>RxDll flop</td> </tr> <tr> <td>10b</td> <td>TxDll flop</td> </tr> <tr> <td>11b</td> <td>TxPad flop</td> </tr> </tbody> </table>	Bits	Description	00b	PRE flop	01b	RxDll flop	10b	TxDll flop	11b	TxPad flop
Bits	Description										
00b	PRE flop										
01b	RxDll flop										
10b	TxDll flop										
11b	TxPad flop										
5:4	<b>TrChipSel: training DIMM select.</b> Read-write. Specifies a timing control for a corresponding rank which is to be trained.If D18F2xA8_dct[0][PerRankTimingEn]=1 then the function is CS. Otherwise the function is DIMM. <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>DIMM/CS 0</td> </tr> <tr> <td>01b</td> <td>DIMM/CS 1</td> </tr> <tr> <td>10b</td> <td>DIMM:Reserved/CS 2</td> </tr> <tr> <td>11b</td> <td>DIMM:Reserved/CS 3</td> </tr> </tbody> </table>	Bits	Description	00b	DIMM/CS 0	01b	DIMM/CS 1	10b	DIMM:Reserved/CS 2	11b	DIMM:Reserved/CS 3
Bits	Description										
00b	DIMM/CS 0										
01b	DIMM/CS 1										
10b	DIMM:Reserved/CS 2										
11b	DIMM:Reserved/CS 3										
3	<b>PhyFenceTrEn: phy fence training enable.</b> Read-write. 1=Initiate phy based fence training. 0=Stop the phy based fence training engine.										
2	<b>TrNibbleSel: training nibble select.</b> Read-write. BIOS: 0. Specifies nibbles of each DIMM data byte trained during write levelization training. 0=Lower nibbles. 1=Upper nibbles.										
1	Reserved.										
0	<b>WrtLvTrEn: write levelization training enable.</b> Read-write. 1=Initiate write levelization (tDQSS margining) training. 0=The phy stops driving DQS and exits write levelization training.										

**D18F2x9C\_x0000\_000B\_dct[0] DRAM Phy Status Register**

Bits	Description
31	<b>DynModeChange: dynamic mode change.</b> RAZ; write. Reset: 0. 1=Phy enters the state specified by PhySelfRefreshMode.
30	<b>PhyPSReq: phy pstate request.</b> RAZ; write. Reset: 0. 1=Phy enters the memory P-state specified by PhyPS.
29:27	Reserved.
26	<b>PhyPS: phy pstate.</b> RAZ; write. Reset: 0. 1=M1. 0=M0. See PhyPSReq.
25:24	Reserved
23	<b>PhySelfRefreshMode: phy self refresh mode.</b> RAZ; write. Reset: 0. 1=Enter self refresh mode. 0=Exit self refresh mode. See DynModeChange.
22:0	Reserved

**D18F2x9C\_x0000\_000C\_dct[0] DRAM Phy Miscellaneous**

Bits	Description
31	Reserved.

30:26	<b>FenceThresholdTxDll: phy fence threshold transmit DLL.</b> Read-write. Cold reset: 13h. BIOS: See <a href="#">2.9.7.4.3</a> . This field specifies the fence delay threshold value used for DQS receiver valid. See FenceThresholdTxPad.																		
25:21	<b>FenceThresholdRxDll: phy fence threshold DQS receiver enable.</b> Read-write. Cold reset: 13h. BIOS: See <a href="#">2.9.7.4.3</a> . This field specifies the fence delay threshold value used for DQS receiver enable. See FenceThresholdTxPad.																		
20:16	<b>FenceThresholdTxPad: phy fence threshold transmit pad.</b> Read-write. Cold reset: 13h. BIOS: See <a href="#">2.9.7.4.3</a> . This field specifies the fence delay threshold value used for write data, write DQS, Addr/Cmd, CS, ODT, and CKE. <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00h</td> <td>0/64 MEMCLK delay</td> </tr> <tr> <td>1Eh-01h</td> <td>&lt;FenceThresholdTxPad&gt;/64 MEMCLK delay</td> </tr> <tr> <td>1Fh</td> <td>31/64 MEMCLK delay</td> </tr> </tbody> </table>	Bits	Description	00h	0/64 MEMCLK delay	1Eh-01h	<FenceThresholdTxPad>/64 MEMCLK delay	1Fh	31/64 MEMCLK delay										
Bits	Description																		
00h	0/64 MEMCLK delay																		
1Eh-01h	<FenceThresholdTxPad>/64 MEMCLK delay																		
1Fh	31/64 MEMCLK delay																		
15:12	<b>CKETri: CKE tri-state.</b> Read-write. Cold reset: 0h. 0=The CKE signal is not tri-stated. 1=Tri-state unconnected CKE signal from the processor. See <a href="#">2.9.4 [DDR Pad to Processor Pin Mapping]</a> . <table border="1"> <thead> <tr> <th>Bit</th> <th>Pad</th> </tr> </thead> <tbody> <tr> <td>[0]</td> <td>MEMCKE[0][0]</td> </tr> <tr> <td>[1]</td> <td>MEMCKE[0][1]</td> </tr> <tr> <td>[2]</td> <td>MEMCKE[0][2]</td> </tr> <tr> <td>[3]</td> <td>MEMCKE[0][3]</td> </tr> </tbody> </table>	Bit	Pad	[0]	MEMCKE[0][0]	[1]	MEMCKE[0][1]	[2]	MEMCKE[0][2]	[3]	MEMCKE[0][3]								
Bit	Pad																		
[0]	MEMCKE[0][0]																		
[1]	MEMCKE[0][1]																		
[2]	MEMCKE[0][2]																		
[3]	MEMCKE[0][3]																		
11:8	<b>ODTTri: ODT tri-state.</b> Read-write. Cold reset: 0h. 0=The ODT signals are not tri-stated unless directed to by the DCT. 1=Tri-state unconnected ODT signals from the processor. See <a href="#">2.9.4 [DDR Pad to Processor Pin Mapping]</a> . <table border="1"> <thead> <tr> <th>Bit</th> <th>Pad</th> </tr> </thead> <tbody> <tr> <td>[0]</td> <td>MEMODT[0][0]</td> </tr> <tr> <td>[1]</td> <td>MEMODT[0][1]</td> </tr> <tr> <td>[2]</td> <td>MEMODT[0][2]</td> </tr> <tr> <td>[3]</td> <td>MEMODT[0][3]</td> </tr> </tbody> </table>	Bit	Pad	[0]	MEMODT[0][0]	[1]	MEMODT[0][1]	[2]	MEMODT[0][2]	[3]	MEMODT[0][3]								
Bit	Pad																		
[0]	MEMODT[0][0]																		
[1]	MEMODT[0][1]																		
[2]	MEMODT[0][2]																		
[3]	MEMODT[0][3]																		
7:0	<b>ChipSelTri: chip select tri-state.</b> Read-write. Cold reset: 00h. 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. See <a href="#">2.9.4 [DDR Pad to Processor Pin Mapping]</a> . <table border="1"> <thead> <tr> <th>Bit</th> <th>Pad</th> </tr> </thead> <tbody> <tr> <td>[0]</td> <td>MEMCS[0]_L[0]</td> </tr> <tr> <td>[1]</td> <td>MEMCS[0]_L[1]</td> </tr> <tr> <td>[2]</td> <td>MEMCS[0]_L[2]</td> </tr> <tr> <td>[3]</td> <td>MEMCS[0]_L[3]</td> </tr> <tr> <td>[4]</td> <td>MEMCS[0]_L[4]</td> </tr> <tr> <td>[5]</td> <td>MEMCS[0]_L[5]</td> </tr> <tr> <td>[6]</td> <td>MEMCS[0]_L[6]</td> </tr> <tr> <td>[7]</td> <td>MEMCS[0]_L[7]</td> </tr> </tbody> </table>	Bit	Pad	[0]	MEMCS[0]_L[0]	[1]	MEMCS[0]_L[1]	[2]	MEMCS[0]_L[2]	[3]	MEMCS[0]_L[3]	[4]	MEMCS[0]_L[4]	[5]	MEMCS[0]_L[5]	[6]	MEMCS[0]_L[6]	[7]	MEMCS[0]_L[7]
Bit	Pad																		
[0]	MEMCS[0]_L[0]																		
[1]	MEMCS[0]_L[1]																		
[2]	MEMCS[0]_L[2]																		
[3]	MEMCS[0]_L[3]																		
[4]	MEMCS[0]_L[4]																		
[5]	MEMCS[0]_L[5]																		
[6]	MEMCS[0]_L[6]																		
[7]	MEMCS[0]_L[7]																		

### **D18F2x9C\_x0000\_000D\_dct[0]\_mp[1:0] DRAM Phy DLL Control**

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). These values are programmed by BIOS based on programmed DDR frequency. This register must be programmed before DRAM device initialization.

Bits	Description
31:26	Reserved.
25:24	<b>RxDLLWakeupTime: receive DLL wakeup time.</b> Read-write. BIOS: See 2.9.7.10. 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	<b>RxCPUpdPeriod: receive charge pump period.</b> Read-write. BIOS: See 2.9.7.10. Specifies the number of DLL relocks required to keep the receive DLLs locked for the period where there is no read traffic.
19:16	<b>RxMaxDurDllNoLock: receive maximum duration DLL no lock.</b> Read-write. BIOS: See 2.9.7.7. Specifies the number of PCLK cycles that occur before the phy DLLs relock. A DLL relock occurs every $2^{\text{RxMaxDurDllNoLock}}$ if there are no reads during the period. 0=DLL power saving disabled.
15:10	Reserved.
9:8	<b>TxDLLWakeupTime: transmit DLL wakeup time.</b> Read-write. BIOS: See 2.9.7.10. 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	<b>TxCPUpdPeriod: transmit charge pump DLL wakeup time.</b> Read-write. BIOS: See 2.9.7.10. Specifies the number of DLL relocks required to keep the TxDLLs locked for the period where there is no write traffic.
3:0	<b>TxMaxDurDllNoLock: transmit maximum duration DLL no lock.</b> Read-write. BIOS: See 2.9.7.7. Specifies the number of PCLK cycles that occur before the phy DLLs relock. A DLL relock occurs every $2^{\text{TxMaxDurDllNoLock}}$ if there are no writes during the period. 0=DLL power saving disabled.

### D18F2x9C\_x0000\_00[2A:10]\_dct[0]\_mp[1:0] DRAM DQS Receiver Enable Timing

**Table 130: Index Mapping for D18F2x9C\_x0000\_00[2A:10]\_dct[0]\_mp[1:0]**

D18F2x98_dct[0][31:0]	Function
0000_0010h	DIMM/CS 0 Bytes 1-0
0000_0011h	DIMM/CS 0 Bytes 3-2
0000_0012h	DIMM/CS 0 ECC
0000_0013h	DIMM /CS 1 Bytes 1-0
0000_0014h	DIMM /CS 1 Bytes 3-2
0000_0015h	DIMM /CS 1 ECC
0000_0016h	Reserved /CS 2 Bytes 1-0
0000_0017h	Reserved /CS 2 Bytes 3-2
0000_0018h	Reserved /CS 2 ECC

**Table 130: Index Mapping for D18F2x9C\_x0000\_00[2A:10]\_dct[0]\_mp[1:0]**

0000_0019h	Reserved /CS 3 Bytes 1-0
0000_001Ah	Reserved /CS 3 Bytes 3-2
0000_001Bh	Reserved /CS 3 ECC
0000_001[F:C]h	Reserved
0000_0020h	DIMM/CS 0 Bytes 5-4
0000_0021h	DIMM/CS 0 Bytes 7-6
0000_0023h	DIMM /CS 1 Bytes 5-4
0000_0024h	DIMM /CS 1 Bytes 7-6
0000_0025h	Reserved
0000_0026h	Reserved /CS 2 Bytes 5-4
0000_0027h	Reserved /CS 2 Bytes 7-6
0000_0029h	Reserved /CS 3 Bytes 5-4
0000_002Ah	Reserved /CS 3 Bytes 7-6

**Table 131: Byte Lane Mapping for D18F2x9C\_x0000\_00[2A:10]\_dct[0]\_mp[1:0]**

Register	Bits	
	25:16	9:0
D18F2x9C_x0000_001[9,6,3,0]_dct[0]_mp[1:0]	Byte1	Byte0
D18F2x9C_x0000_001[A,7,4,1]_dct[0]_mp[1:0]	Byte3	Byte2
D18F2x9C_x0000_001[B,8,5,2]_dct[0]_mp[1:0]	Reserved	ECC
D18F2x9C_x0000_002[9,6,3,0]_dct[0]_mp[1:0]	Byte5	Byte4
D18F2x9C_x0000_002[A,7,4,1]_dct[0]_mp[1:0]	Byte7	Byte6

If **D18F2xA8\_dct[0][PerRankTimingEn]=1** then the function is CS. Otherwise the function is DIMM.

Each of these registers control the timing of the receiver enable from the start of the read preamble with respect to MEMCLK. See 2.9.7.9 [DRAM Training]. These delay registers must be programmed such that across all DIMMs and lanes  $\text{MAX}(\text{DqsRcvEnDelay}) - \text{MIN}(\text{DqsRcvEnDelay}) \leq 7 \text{ UI}$ .

Bits	Description
31:26	Reserved.
25:21	<b>DqsRcvEnGrossDelay: DQS receiver enable gross delay.</b> See: <a href="#">D18F2x9C_x0000_00[2A:10]_dct[0]_mp[1:0][9:5]</a> .
20:16	<b>DqsRcvEnFineDelay: DQS receiver enable fine delay.</b> See: <a href="#">D18F2x9C_x0000_00[2A:10]_dct[0]_mp[1:0][4:0]</a> .
15:10	Reserved.

9:5	<b>DqsRcvEnGrossDelay: DQS receiver enable gross delay.</b> Read-write. Reset: 01h.	
	<u>Bits</u>	<u>Description</u>
	00h	No delay
	01h	0 MEMCLK delay
	1Eh-02h	<DqsRcvEnGrossDelay>/2 MEMCLK delay
1Fh	15.5 MEMCLK delay	
4:0	<b>DqsRcvEnFineDelay: DQS receiver enable fine delay.</b> Read-write. Cold reset: 00h.	
	<u>Bits</u>	<u>Description</u>
	00h	0/64 MEMCLK delay
	1Eh-01h	<DqsRcvEnFineDelay>/64 MEMCLK delay
	1Fh	31/64 MEMCLK delay

**D18F2x9C\_x0000\_00[4A:30]\_dct[0]\_mp[1:0] DRAM DQS Write Timing**

BIOS: See 2.9.7.9 [DRAM Training].

**Table 132: Index Mapping for D18F2x9C\_x0000\_00[4A:30]\_dct[0]\_mp[1:0]**

D18F2x98_dct[0][31:0]	Function
0000_0030h	DIMM/CS 0 Bytes 1-0
0000_0031h	DIMM/CS 0 Bytes 3-2
0000_0032h	DIMM/CS 0 ECC
0000_0033h	DIMM /CS 1 Bytes 1-0
0000_0034h	DIMM /CS 1 Bytes 3-2
0000_0035h	DIMM /CS 1 ECC
0000_0036h	Reserved /CS 2 Bytes 1-0
0000_0037h	Reserved /CS 2 Bytes 3-2
0000_0038h	Reserved /CS 2 ECC
0000_0039h	Reserved /CS 3 Bytes 1-0
0000_003Ah	Reserved /CS 3 Bytes 3-2
0000_003Bh	Reserved /CS 3 ECC
0000_0040h	DIMM/CS 0 Bytes 5-4
0000_0041h	DIMM/CS 0 Bytes 7-6
0000_0043h	DIMM /CS 1 Bytes 5-4
0000_0044h	DIMM /CS 1 Bytes 7-6
0000_0046h	Reserved /CS 2 Bytes 5-4

**Table 132: Index Mapping for D18F2x9C\_x0000\_00[4A:30]\_dct[0]\_mp[1:0]**

0000_0047h	Reserved /CS 2 Bytes 7-6
0000_0049h	Reserved /CS 3 Bytes 5-4
0000_004Ah	Reserved /CS 3 Bytes 7-6

**Table 133: Byte Lane Mapping for D18F2x9C\_x0000\_00[4A:30]\_dct[0]\_mp[1:0]**

Register	Bits	
	23:16	7:0
D18F2x9C_x0000_003[9,6,3,0]_dct[0]_mp[1:0]	Byte1	Byte0
D18F2x9C_x0000_003[A,7,4,1]_dct[0]_mp[1:0]	Byte3	Byte2
D18F2x9C_x0000_003[B,8,5,2]_dct[0]_mp[1:0]	Reserved	ECC
D18F2x9C_x0000_004[9,6,3,0]_dct[0]_mp[1:0]	Byte5	Byte4
D18F2x9C_x0000_004[A,7,4,1]_dct[0]_mp[1:0]	Byte7	Byte6

If **D18F2xA8\_dct[0][PerRankTimingEn]=1** then the function is CS. Otherwise the function is DIMM. Each of these registers control the DQS timing delay for write commands relative to MEMCLK. The delay starts at the rise edge of MEMCLK corresponding to the CAS-writelatency. Each control includes a gross timing field and a fine timing field, the sum of which is the total delay. See 2.9.7.9 [DRAM Training].

Bits	Description										
31:29	Reserved.										
28:24	Reserved.										
23:21	<b>WrDqsGrossDly: DQS write gross delay.</b> See: D18F2x9C_x0000_00[4A:30]_dct[0]_mp[1:0][7:5].										
20:16	<b>WrDqsFineDly: DQS write fine delay.</b> See: D18F2x9C_x0000_00[4A:30]_dct[0]_mp[1:0][4:0].										
15:13	Reserved.										
12:8	Reserved.										
7:5	<b>WrDqsGrossDly: DQS write gross delay.</b> Read-write. Reset: 0. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>No delay</td> </tr> <tr> <td>001b</td> <td>0.5 MEMCLK delay</td> </tr> <tr> <td>110b-010b</td> <td>&lt;WrDqsGrossDly&gt;/2 MEMCLK delay</td> </tr> <tr> <td>111b</td> <td>3.5 MEMCLK delay</td> </tr> </tbody> </table>	Bits	Description	000b	No delay	001b	0.5 MEMCLK delay	110b-010b	<WrDqsGrossDly>/2 MEMCLK delay	111b	3.5 MEMCLK delay
Bits	Description										
000b	No delay										
001b	0.5 MEMCLK delay										
110b-010b	<WrDqsGrossDly>/2 MEMCLK delay										
111b	3.5 MEMCLK delay										
4:0	<b>WrDqsFineDly: DQS write fine delay.</b> Read-write. Cold reset: 0. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00h</td> <td>0/64 MEMCLK delay</td> </tr> <tr> <td>1Eh-01h</td> <td>&lt;WrDqsFineDly&gt;/64 MEMCLK delay</td> </tr> <tr> <td>1Fh</td> <td>31/64 MEMCLK delay</td> </tr> </tbody> </table>	Bits	Description	00h	0/64 MEMCLK delay	1Eh-01h	<WrDqsFineDly>/64 MEMCLK delay	1Fh	31/64 MEMCLK delay		
Bits	Description										
00h	0/64 MEMCLK delay										
1Eh-01h	<WrDqsFineDly>/64 MEMCLK delay										
1Fh	31/64 MEMCLK delay										

**D18F2x9C\_x0000\_00[52:50]\_dct[0] DRAM Phase Recovery Control**

These registers are used by BIOS 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. See 2.9.7.9 [DRAM Training].

**Table 134: Index Mapping for D18F2x9C\_x0000\_00[52:50]\_dct[0]**

D18F2x98_dct[0][31:0]	Function
0000_0050h	Bytes 3-0
0000_0051h	Bytes 7-4
0000_0052h	ECC

**Table 135: Byte Lane Mapping for D18F2x9C\_x0000\_00[52:50]\_dct[0]**

Register	Bits			
	31:24	23:16	15:8	7:0
D18F2x9C_x0000_0050_dct[0]	Byte3	Byte2	Byte1	Byte0
D18F2x9C_x0000_0051_dct[0]	Byte7	Byte6	Byte5	Byte4
D18F2x9C_x0000_0052_dct[0]	Reserved	Reserved	Reserved	ECC

Bits	Description										
31	Reserved.										
30:29	<b>PhRecGrossDly: phase recovery gross delay.</b> See: PhRecGrossDly.										
28:24	<b>PhRecFineDly: phase recovery fine delay.</b> See: PhRecFineDly.										
23	Reserved.										
22:21	<b>PhRecGrossDly: phase recovery gross delay.</b> See: PhRecGrossDly.										
20:16	<b>PhRecFineDly: phase recovery fine delay.</b> See: PhRecFineDly.										
15	Reserved.										
14:13	<b>PhRecGrossDly: phase recovery gross delay.</b> See: PhRecGrossDly.										
12:8	<b>PhRecFineDly: phase recovery fine delay.</b> See: PhRecFineDly.										
7	Reserved.										
6:5	<b>PhRecGrossDly: phase recovery gross delay.</b> Read-write. Reset: X. Gross timing indicates the number of half-MEMCLK periods that the phase recovery engine advanced while aligning edges. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>No delay</td> </tr> <tr> <td>01b</td> <td>0.5 MEMCLK delay</td> </tr> <tr> <td>10b</td> <td>1.0 MEMCLK delay</td> </tr> <tr> <td>11b</td> <td>1.5 MEMCLK delay</td> </tr> </tbody> </table>	Bits	Description	00b	No delay	01b	0.5 MEMCLK delay	10b	1.0 MEMCLK delay	11b	1.5 MEMCLK delay
Bits	Description										
00b	No delay										
01b	0.5 MEMCLK delay										
10b	1.0 MEMCLK delay										
11b	1.5 MEMCLK delay										
4:0	<b>PhRecFineDly: phase recovery fine delay.</b> Read-write. Reset: X. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00h</td> <td>0/64 MEMCLK delay</td> </tr> <tr> <td>1Eh-01h</td> <td>&lt;PhRecFineDly&gt;/64 MEMCLK delay</td> </tr> <tr> <td>1Fh</td> <td>31/64 MEMCLK delay</td> </tr> </tbody> </table>	Bits	Description	00h	0/64 MEMCLK delay	1Eh-01h	<PhRecFineDly>/64 MEMCLK delay	1Fh	31/64 MEMCLK delay		
Bits	Description										
00h	0/64 MEMCLK delay										
1Eh-01h	<PhRecFineDly>/64 MEMCLK delay										
1Fh	31/64 MEMCLK delay										

**D18F2x9C\_x0D0F\_0[F,8:0]02\_dct[0] Data Byte Transmit PreDriver Calibration**

Cold reset: xxxx\_xxxxh. BIOS: See 2.9.7.4.4.

Table 136: Index Mapping for D18F2x9C\_x0D0F\_0[F,8:0]02\_dct[0]

D18F2x98_dct[0][31:16]	D18F2x98_dct[0][15:0]								
	0802h	0702h	0602h	0502h	0402h	0302h	0202h	0102h	0002h
0D0Fh	ECC	Byte 7	Byte 6	Byte 5	Byte 4	Byte 3	Byte 2	Byte 1	Byte 0

Table 137: Broadcast Mapping for D18F2x9C\_x0D0F\_0[F,8:0]02\_dct[0]

D18F2x98_dct[0][31:16]	D18F2x98_dct[0][15:0]
	0F02h
0D0Fh	D18F2x9C_x0D0F_0[8:0]02

Table 138: Valid Values for D18F2x9C\_x0D0F\_0[F,8:0]02\_dct[0][TxPreP]

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	<b>ValidTxAndPre: predriver calibration code valid.</b> Read-write; cleared-by-hardware. 1=Predriver calibration codes are copied from this register and D18F2x9C_x0D0F_0[F,8:0]0[A,6]_dct[0] into the associated transmit pad. Hardware clears this field after the copy is complete.
14:12	Reserved.
11:6	<b>TxPreP: PMOS predriver calibration code.</b> Read-write. Specifies the rising edge slew rate of the transmit pad. See: Table 138 [Valid Values for D18F2x9C_x0D0F_0[F,8:0]02_dct[0][TxPreP]]. After updating this value, BIOS must program ValidTxAndPre=1 for the change to take effect.
5:0	<b>TxPreN: NMOS predriver calibration code.</b> Read-write. Specifies the falling edge slew rate of the transmit pad. See: Table 138 [Valid Values for D18F2x9C_x0D0F_0[F,8:0]02_dct[0][TxPreP]]. After updating this value, BIOS must program ValidTxAndPre=1 for the change to take effect.



**D18F2x9C\_x0D0F\_0[F,8:0]0[B,7,3]\_dct[0] Data Byte P0dt Configuration**Cold reset: 0000\_xxxxh. See [D18F2x9C\\_x0D0F\\_1C00\\_dct\[0\]](#).**Table 139: Index Mapping for D18F2x9C\_x0D0F\_0[F,8:0]0[B,7,3]\_dct[0]**

D18F2x98_dct[0][31:0]	Function
0D0F_000[B,7,3]h	Byte 0
0D0F_010[B,7,3]h	Byte 1
0D0F_020[B,7,3]h	Byte 2
0D0F_030[B,7,3]h	Byte 3
0D0F_040[B,7,3]h	Byte 4
0D0F_050[B,7,3]h	Byte 5
0D0F_060[B,7,3]h	Byte 6
0D0F_070[B,7,3]h	Byte 7
0D0F_080[B,7,3]h	ECC
0D0F_0F0[B,7,3]h	D18F2x9C_x0D0F_0[8:0]0[B,7,3]

Bits	Description
31:16	Reserved.
15:14	Reserved.
13:8	<b>P0dtD: pmos ODT calibration data.</b> Read-write.
7:6	Reserved.
5:0	<b>N0dtD: nmos ODT calibration data.</b> Read-write.

**D18F2x9C\_x0D0F\_0[F,8:0]04\_dct[0]\_mp[1:0] Data Byte Pad Configuration**

Cold reset: 0000\_0033h.

**Table 140: Index Mapping for D18F2x9C\_x0D0F\_0[F,8:0]04\_dct[0]\_mp[1:0]**

D18F2x98_dct[0][31:0]	Function
0D0F_0004h	Byte 0
0D0F_0104h	Byte 1
0D0F_0204h	Byte 2
0D0F_0304h	Byte 3
0D0F_0404h	Byte 4
0D0F_0504h	Byte 5
0D0F_0604h	Byte 6
0D0F_0704h	Byte 7
0D0F_0804h	ECC
0D0F_0F04h	D18F2x9C_x0D0F_0[8:0]04

Bits	Description
31:14	Reserved.

13	<b>TriDM: tri-state DM.</b> Read-write. BIOS: 2.9.7.10. Read-write. Specifies tri-state control for the memory DM signal. 1=Signal is tri-stated. 0=Signal is not tri-stated.
12	<b>POdtOff: processor ODT off.</b> Read-write. BIOS: See 2.9.7.6.6.1=Phy disables receiver pad termination. 0=Phy enables receiver pad termination. Hardware programs this field for the current M-state (D18F2x9C_x0D00_E008_dct[0][PhyPS]) with the value in D18F2x94_dct[0][ProcOdtDis] when the memory frequency is updated. See 2.9.7.4.2. BIOS must reprogram this field after each frequency change if the target value differs from D18F2x94_dct[0][ProcOdtDis].
11:0	Reserved.

**D18F2x9C\_x0D0F\_0[F,8:0]0[A,6]\_dct[0] Data Byte Transmit PreDriver Calibration 2**

Cold reset: xxxx\_xxxxh. BIOS: See 2.9.7.4.4.

Table 141: Index Mapping for D18F2x9C\_x0D0F\_0[F,8:0]06\_dct[0]

D18F2x98_dct[0][31:16]	D18F2x98_dct[0][15:0]								
	0806h	0706h	0606h	0506h	0406h	0306h	0206h	0106h	0006h
0D0Fh	ECC	Byte 7	Byte 6	Byte 5	Byte 4	Byte 3	Byte 2	Byte 1	Byte 0

Table 142: Index Mapping for D18F2x9C\_x0D0F\_0[F,8:0]0A\_dct[0]

D18F2x98_dct[0][31:16]	D18F2x98_dct[0][15:0]								
	080Ah	070Ah	060Ah	050Ah	040Ah	030Ah	020Ah	010Ah	000Ah
0D0Fh	ECC	Byte 7	Byte 6	Byte 5	Byte 4	Byte 3	Byte 2	Byte 1	Byte 0

Table 143: Broadcast Mapping for D18F2x9C\_x0D0F\_0[F,8:0]0[A,6]\_dct[0]

D18F2x98_dct[0][31:16]	D18F2x98_dct[0][15:0]	
	0F0Ah	0F06h
0D0Fh	D18F2x9C_x0D0F_0[8:0]0A	D18F2x9C_x0D0F_0[8:0]06

Bits	Description
31:12	Reserved.
11:6	<b>TxPreP: PMOS predriver calibration code.</b> Read-write. This field specifies the rising edge slew rate of the transmit pad. See: D18F2x9C_x0D0F_0[F,8:0]02_dct[0][TxPreP]. After updating this value, BIOS must program D18F2x9C_x0D0F_0[F,8:0]02_dct[0][ValidTxAndPre]=1 for the change to take effect.
5:0	<b>TxPreN: NMOS predriver calibration code.</b> Read-write. This field specifies the falling edge slew rate of the transmit pad. See: D18F2x9C_x0D0F_0[F,8:0]02_dct[0][TxPreN]. After updating this value, BIOS must program D18F2x9C_x0D0F_0[F,8:0]02_dct[0][ValidTxAndPre]=1 for the change to take effect.

**D18F2x9C\_x0D0F\_0[F,8:0]10\_dct[0]\_mp[1:0] Data Byte DLL Power Management**

Cold reset: 0000\_0000h.

**Table 144: Index Mapping for D18F2x9C\_x0D0F\_0[F,8:0]10\_dct[0]\_mp[1:0]**

D18F2x98_dct[0][31:0]	Function
0D0F_0010h	Byte 0
0D0F_0110h	Byte 1
0D0F_0210h	Byte 2
0D0F_0310h	Byte 3
0D0F_0410h	Byte 4
0D0F_0510h	Byte 5
0D0F_0610h	Byte 6
0D0F_0710h	Byte 7
0D0F_0810h	Byte 8
0D0F_0F10h	Byte 8-0

Bits	Description
31:13	Reserved.
12	<b>EnRxPadStandby: enable receiver pad standby.</b> Read-write. BIOS: See 2.9.7.7.1=Phy receiver standby mode is enabled to save power when not receiving data. 0=Phy receiver standby mode is disabled.
11:0	Reserved.

**D18F2x9C\_x0D0F\_0[F,8:0]13\_dct[0]\_mp[1:0] Data Byte DLL Configuration****Table 145: Index Mapping for D18F2x9C\_x0D0F\_0[F,8:0]13\_dct[0]\_mp[1:0]**

D18F2x98_dct[0][31:0]	Function
0D0F_0013h	Byte 0
0D0F_0113h	Byte 1
0D0F_0213h	Byte 2
0D0F_0313h	Byte 3
0D0F_0413h	Byte 4
0D0F_0513h	Byte 5
0D0F_0613h	Byte 6
0D0F_0713h	Byte 7
0D0F_0813h	Byte 8
0D0F_0F13h	Byte 8-0

Bits	Description
31:15	Reserved.
14	<b>ProcOdtAdv: ProcOdt advance.</b> Read-write. Cold reset: 1. BIOS: IF ((Solder-down DRAM    SODIMM) && DdrRate <= 1333) THEN 0 ELSE 1 ENDIF. 0=ProcOdt is asserted 1.5 PCLK before DqsEn . 1=If preceding write, ProcOdt is asserted 2.5 PCLK before DqsEn, else, ProcOdt is asserted 5.0 PCLK before DqsEn.

13	<b>ExtraProcOdtAdv: extra ProcOdt advance.</b> Read-write. Cold reset: 0. BIOS: 0. Reserved if ProcOdtAdv=0. 1=For a read without preceding read or write commands, ProcOdt is asserted with the CAS command . 0=Extra ProcOdt is disabled.
12:9	Reserved.
8	<b>RxSsbMntClkEn: receive channel maintenance clock enable.</b> Read-write. Cold reset: 0. BIOS: 2.9.7.10. 1=Enable receive channel maintenance clocks to improve internal timing margin at the cost of some extra power. To enable clock generation, BIOS must first enable maintenance clocks in the DCT (see D18F2x248_dct[0]_mp[1:0][RxChMntClkEn]). 0=Disable clocks. The maintenance clock period is fixed at 8 PCLKs when no traffic occurs.
7:2	Reserved.
1	<b>DllDisEarlyU: DLL disable early upper.</b> Read-write. Cold reset: 0. BIOS: See 2.9.7.10. 1=Disable upper receiver DQS DLL early timing for power savings.
0	<b>DllDisEarlyL: DLL disable early lower.</b> Read-write. Cold reset: 0. BIOS: See 2.9.7.10. 1=Disable lower receiver DQS DLL early timing for power savings.

**D18F2x9C\_x0D0F\_0[F,8:0]1C\_dct[0]\_mp[1:0] Data Byte DLL Power Management**

Cold reset: 0000\_0000h.

**Table 146: Index Mapping for D18F2x9C\_x0D0F\_0[F,8:0]1C\_dct[0]\_mp[1:0]**

D18F2x98_dct[0][31:0]	Function
0D0F_001Ch	Byte 0
0D0F_011Ch	Byte 1
0D0F_021Ch	Byte 2
0D0F_031Ch	Byte 3
0D0F_041Ch	Byte 4
0D0F_051Ch	Byte 5
0D0F_061Ch	Byte 6
0D0F_071Ch	Byte 7
0D0F_081Ch	Byte 8
0D0F_0F1Ch	Byte 8-0

Bits	Description
15	<b>RxDllStggrEn: Rx DLL stagger enable.</b> Read-write. BIOS: 2.9.7.10. 1=Insert RxDllStggrDly delay prior to waking the receive DLLs from standby. 0=DLLs wake from standby at the default pending-activity reference point.
14	Reserved.
13:8	<b>RxDllStggrDly[5:0]: Rx DLL stagger delay.</b> Read-write. BIOS: 2.9.7.10. Specifies the delay in PCLK cycles from a CAS command or an internal DLL maintenance lock timing event before the receive DLLs exit standby if the data bus is still idle. If data bus traffic is pending and the DLLs are in standby then they wake immediately regardless of the delay value.
7	<b>TxDllStggrEn: Tx DLL stagger enable.</b> Read-write. BIOS: 2.9.7.10. 1=Insert TxDllStggrDly delay prior to waking the transmit DLL from standby. 0=DLL wakes from standby at the default pending-activity reference point.

6	Reserved.
5:0	<b>TxDllStggrDly[5:0]: Tx DLL stagger delay.</b> Read-write. BIOS: 0. BIOS: 2.9.7.10. Specifies the delay in PCLK cycles from a CAS command or an internal DLL maintenance lock timing event before the transmit DLL exits standby if the data bus is still idle. If data bus traffic is pending and the DLL is in standby then it wakes immediately regardless of the delay value.

**D18F2x9C\_x0D0F\_0[F,8:0]1E\_dct[0]\_mp[1:0] Data Byte Receiver Bias Current Control**

Cold reset: 0000\_5220h.

**Table 147: Index Mapping for D18F2x9C\_x0D0F\_0[F,8:0]1E\_dct[0]\_mp[1:0]**

D18F2x98_dct[0][31:0]	Function
0D0F_001Eh	Byte 0
0D0F_011Eh	Byte 1
0D0F_021Eh	Byte 2
0D0F_031Eh	Byte 3
0D0F_041Eh	Byte 4
0D0F_051Eh	Byte 5
0D0F_061Eh	Byte 6
0D0F_071Eh	Byte 7
0D0F_081Eh	Byte 8
0D0F_0F1Eh	Byte 8-0

Bits	Description																		
31:15	Reserved.																		
14:12	<b>DllCSRBIASTrim: dll csr biast trim.</b> Read-write. BIOS: 001b. The MSB bit adjusts the DAC current profile that is used only by the delayline, which adjusts the tuning curve of the delayline. The remaining two bits affect the global bias current. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>-25%</td> </tr> <tr> <td>001b</td> <td>Nominal</td> </tr> <tr> <td>010b</td> <td>+50%</td> </tr> <tr> <td>011b</td> <td>+100</td> </tr> <tr> <td>100b</td> <td>-25% + delayline adjust</td> </tr> <tr> <td>101b</td> <td>Nominal + delayline adjust</td> </tr> <tr> <td>110b</td> <td>+50% + delayline adjust</td> </tr> <tr> <td>111b</td> <td>+100 + delayline adjust</td> </tr> </tbody> </table>	Bits	Description	000b	-25%	001b	Nominal	010b	+50%	011b	+100	100b	-25% + delayline adjust	101b	Nominal + delayline adjust	110b	+50% + delayline adjust	111b	+100 + delayline adjust
Bits	Description																		
000b	-25%																		
001b	Nominal																		
010b	+50%																		
011b	+100																		
100b	-25% + delayline adjust																		
101b	Nominal + delayline adjust																		
110b	+50% + delayline adjust																		
111b	+100 + delayline adjust																		
11:8	Reserved.																		
7	Reserved.																		
6:0	Reserved.																		

**D18F2x9C\_x0D0F\_0[F,8:0]1F\_dct[0]\_mp[1:0] Data Byte Receiver Configuration**

Cold reset: 0000\_2003h.

Table 148: Index Mapping for D18F2x9C\_x0D0F\_0[F,8:0]1F\_dct[0]\_mp[1:0]

D18F2x98_dct[0][31:0]	Function
0D0F_001Fh	Byte 0
0D0F_011Fh	Byte 1
0D0F_021Fh	Byte 2
0D0F_031Fh	Byte 3
0D0F_041Fh	Byte 4
0D0F_051Fh	Byte 5
0D0F_061Fh	Byte 6
0D0F_071Fh	Byte 7
0D0F_081Fh	Byte 8
0D0F_0F1Fh	Byte 8-0

Bits	Description												
31:8	Reserved.												
7:5	<p><b>RxDqInsDly: receive DQ insertion delay.</b> Read-write. BIOS: 2.9.7.9.4. This control adds insertion delay to the Rx DQ path so that the delay through Rx Dqs+DLL insertion delay+Clk distribution better matches the DQ data propagation delay.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>Min delay</td> </tr> <tr> <td>001b</td> <td>Delay 1</td> </tr> <tr> <td>010b</td> <td>Delay 2</td> </tr> <tr> <td>011b</td> <td>Max delay</td> </tr> <tr> <td>111b-100b</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Description	000b	Min delay	001b	Delay 1	010b	Delay 2	011b	Max delay	111b-100b	Reserved
Bits	Description												
000b	Min delay												
001b	Delay 1												
010b	Delay 2												
011b	Max delay												
111b-100b	Reserved												
4:3	<p><b>RxVioLvl: receiver voltage level.</b> Read-write. BIOS: 2.9.7.4.1. Specifies the VDDIO voltage level.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>1.5 V</td> </tr> <tr> <td>01b</td> <td>1.35 V</td> </tr> <tr> <td>10b</td> <td>1.25 V</td> </tr> <tr> <td>11b</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Description	00b	1.5 V	01b	1.35 V	10b	1.25 V	11b	Reserved		
Bits	Description												
00b	1.5 V												
01b	1.35 V												
10b	1.25 V												
11b	Reserved												
2:0	Reserved.												

#### **D18F2x9C\_x0D0F\_0[F,8:0]2[3:0]\_dct[0]\_mp[1:0] Data Byte RxDqs DLL Configuration**

Cold reset: 0000\_0F0Fh. See D18F2x9C\_x0000\_0[3:0]0[7:5]\_dct[0]\_mp[1:0].

Table 149: Index Mapping for D18F2x9C\_x0D0F\_0[F,8:0]2[3:0]\_dct[0]\_mp[1:0]

D18F2x98_dct[0][31:0]	Function
0D0F_0020h	DIMM/CS 0 Byte 0
0D0F_0120h	DIMM/CS 0 Byte 1
0D0F_0220h	DIMM/CS 0 Byte 2
0D0F_0320h	DIMM/CS 0 Byte 3
0D0F_0420h	DIMM/CS 0 Byte 4
0D0F_0520h	DIMM/CS 0 Byte 5

**Table 149: Index Mapping for D18F2x9C\_x0D0F\_0[F,8:0]2[3:0]\_dct[0]\_mp[1:0]**

0D0F_0620h	DIMM/CS 0 Byte 6
0D0F_0720h	DIMM/CS 0 Byte 7
0D0F_0820h	DIMM/CS 0 Byte 8
0D0F_0F20h	DIMM/CS 0 Byte 8-0
0D0F_0021h	DIMM/CS 1 Byte 0
0D0F_0121h	DIMM/CS 1 Byte 1
0D0F_0221h	DIMM/CS 1 Byte 2
0D0F_0321h	DIMM/CS 1 Byte 3
0D0F_0421h	DIMM/CS 1 Byte 4
0D0F_0521h	DIMM/CS 1 Byte 5
0D0F_0621h	DIMM/CS 1 Byte 6
0D0F_0721h	DIMM/CS 1 Byte 7
0D0F_0821h	DIMM/CS 1 Byte 8
0D0F_0F21h	DIMM/CS 1 Byte 8-0
0D0F_0022h	Reserved/CS 2 Byte 0
0D0F_0122h	Reserved/CS 2 Byte 1
0D0F_0222h	Reserved/CS 2 Byte 2
0D0F_0322h	Reserved/CS 2 Byte 3
0D0F_0422h	Reserved/CS 2 Byte 4
0D0F_0522h	Reserved/CS 2 Byte 5
0D0F_0622h	Reserved/CS 2 Byte 6
0D0F_0722h	Reserved/CS 2 Byte 7
0D0F_0822h	Reserved/CS 2 Byte 8
0D0F_0F22h	Reserved/CS 2 Byte 8-0
0D0F_0023h	Reserved/CS 3 Byte 0
0D0F_0123h	Reserved/CS 3 Byte 1
0D0F_0223h	Reserved/CS 3 Byte 2
0D0F_0323h	Reserved/CS 3 Byte 3
0D0F_0423h	Reserved/CS 3 Byte 4
0D0F_0523h	Reserved/CS 3 Byte 5
0D0F_0623h	Reserved/CS 3 Byte 6
0D0F_0723h	Reserved/CS 3 Byte 7
0D0F_0823h	Reserved/CS 3 Byte 8
0D0F_0F23h	Reserved/CS 3 Byte 8-0

If `D18F2xA8_dct[0][PerRankTimingEn]=1` then the function is CS. Otherwise the function is DIMM.

Bits	Description
31:13	Reserved.
12:8	<b>RdDqsTimeU: read DQS timing control upper nibble.</b> Read-write.
<u>Bits</u> 1Fh-00h	<u>Description</u> <RdDqsTimeU>/64 MEMCLK delay

7:5	Reserved.
4:0	<b>RdDqsTimeL: read DQS timing control lower nibble.</b> Read-write. <u>Bits</u> <u>Description</u> 1Fh-00h                      <RdDqsTimeL>/64 MEMCLK delay

**D18F2x9C\_x0D0F\_0[8:0]2[9:4]\_dct[0]\_mp[1:0] Data Byte DLL Configuration**

Cold reset: 0000\_0000h. Updated by hardware during 2.9.7.4.3 [Phy Fence Programming].

Bits	Description
31:16	Reserved.
15	<b>FenceBitO: fence bit odd.</b> Read-write.
14	<b>Fence2BitO: fence2 bit odd.</b> Read-write.
13:8	Reserved.
7	<b>FenceBitE: fence bit even.</b> Read-write.
6	<b>Fence2BitE: fence2 bit even.</b> Read-write.
5:0	Reserved.

**D18F2x9C\_x0D0F\_0[F,8:0]30\_dct[0] Data Byte DLL Configuration and Power Down**

Cold reset: 0000\_0001h.

**Table 150: Index Addresses for D18F2x9C\_x0D0F\_0[F,8:0]30\_dct[0]**

D18F2x98_dct[0][31:16]	D18F2x98_dct[0][15:0]								
	0830h	0730h	0630h	0530h	0430h	0330h	0230h	0130h	0030h
0D0Fh	ECC	Byte 7	Byte 6	Byte 5	Byte 4	Byte 3	Byte 2	Byte 1	Byte 0

**Table 151: Broadcast Write Index Address for D18F2x9C\_x0D0F\_0[F,8:0]30\_dct[0]**

D18F2x98_dct[0][31:16]	D18F2x98_dct[0][15:0]
	0F30h
0D0Fh	D18F2x9C_x0D0F_0[8:0]30

Bits	Description
31:10	Reserved.
9	<b>TxPclkGateEn: tx pclk gating enable.</b> Read-write. BIOS: 2.9.7.10. 1=DQS TX data pipe PCLK gating is enabled. 0=PCLK gating disabled.
8	<b>BlockRxDqsLock: block rx dqs lock.</b> Read-write. BIOS: 2.9.7.9.3. Specifies how the receive DLLs lock. 1=Lock on PCLK. 0=Lock on both PCLK and the received DQS
7	<b>PchgPdPclkGateEn: precharge powerdown pclk gating enable.</b> Read-write. BIOS: 2.9.7.10. 1=DByte datapipe PCLK gating is enabled. 0=PCLK gating disabled.
6	<b>DataCtlPipePclkGateEn: data control pipe pclk gating enable.</b> Read-write. BIOS: 2.9.7.10. 1=Data control pipe PCLK gating is enabled. 0=PCLK gating disabled.
5	Reserved.



4	<b>PwrDn: power down.</b> Read-write. BIOS: 2.9.7.10. 1=Turn off DLL circuitry.
3:0	Reserved.

**D18F2x9C\_x0D0F\_0[F,8:0]31\_dct[0] Data Byte Fence2 Threshold**

BIOS: 2.9.7.4.3.

Table 152: Index addresses for D18F2x9C\_x0D0F\_0[F,8:0]31\_dct[0]

D18F2x98_dct[0][31:16]	D18F2x98_dct[0][15:0]								
	0831h	0731h	0631h	0531h	0431h	0331h	0231h	0131h	0031h
0D0Fh	ECC	Byte 7	Byte 6	Byte 5	Byte 4	Byte 3	Byte 2	Byte 1	Byte 0

**Table 153:** Broadcast write index address for D18F2x9C\_x0D0F\_0[F,8:0]31\_dct[0]

D18F2x98_dct[0][31:16]	D18F2x98_dct[0][15:0]
	0F31h
0D0Fh	D18F2x9C_x0D0F_0[8:0]31

Bits	Description										
31:10	Reserved.										
9	<b>Fence2EnableTxDll: phy fence2 enable transmit DLL.</b> Read-write. Cold reset: 0. 1=Enable the use of Fence2ThresholdTxDll for transmit DLL fence threshold. 0=Use D18F2x9C_x0000_000C_dct[0][FenceThresholdTxDll].										
8:5	<b>Fence2ThresholdTxDll: phy fence2 threshold transmit DLL.</b> 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	<b>Fence2EnableTxPad: fence2 enable transmit pad.</b> Read-write. Cold reset: 0. 1=Enable the use of Fence2ThresholdTxPad for transmit pad fence threshold. 0=Use D18F2x9C_x0000_000C_dct[0][FenceThresholdTxPad].										
3:0	<b>Fence2ThresholdTxPad: phy fence2 threshold transmit pad.</b> Read-write. Cold reset: 0. If Fence2EnableTxPad=1, this field specifies the fence delay threshold value used for write data and write DQS. <table border="0" style="margin-left: 20px;"> <tr> <td><u>Bits</u></td> <td><u>Definition</u></td> </tr> <tr> <td>0h</td> <td>No delay</td> </tr> <tr> <td>1h</td> <td>1/64 MEMCLK delay</td> </tr> <tr> <td>Eh-2h</td> <td>&lt;Fence2ThresholdTxPad&gt; MEMCLK delay</td> </tr> <tr> <td>Fh</td> <td>15/64 MEMCLK delay</td> </tr> </table>	<u>Bits</u>	<u>Definition</u>	0h	No delay	1h	1/64 MEMCLK delay	Eh-2h	<Fence2ThresholdTxPad> MEMCLK delay	Fh	15/64 MEMCLK delay
<u>Bits</u>	<u>Definition</u>										
0h	No delay										
1h	1/64 MEMCLK delay										
Eh-2h	<Fence2ThresholdTxPad> MEMCLK delay										
Fh	15/64 MEMCLK delay										

**D18F2x9C\_x0D0F\_0[F,8:0]38\_dct[0] Data Byte DLL Control Register**

Table 154: Index addresses for D18F2x9C\_x0D0F\_0[F,8:0]38\_dct[0]

D18F2x98_dct[0][31:16]	D18F2x98_dct[0][15:0]								
	0838h	0738h	0638h	0538h	0438h	0338h	0238h	0138h	0038h
0D0Fh	ECC	Byte 7	Byte 6	Byte 5	Byte 4	Byte 3	Byte 2	Byte 1	Byte 0

**Table 155:** Broadcast write index address for D18F2x9C\_x0D0F\_0[F,8:0]38\_dct[0]

D18F2x98_dct[0][31:16]	D18F2x98_dct[0][15:0]
	0F38h
0D0Fh	D18F2x9C_x0D0F_0[8:0]38

Bits	Description										
31:16	Reserved.										
15	Reserved.										
14:13	<b>ReducedLoop: reduced loop.</b> Read-write. Cold reset: 0. Specifies additional delay imposed before the phase comparator, which has the effect of locking the DLL to a shorter period. <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>No delay.</td> </tr> <tr> <td>01b</td> <td>1/3 maximum additional delay</td> </tr> <tr> <td>10b</td> <td>2/3 maximum additional delay</td> </tr> <tr> <td>11b</td> <td>Maximum additional delay</td> </tr> </tbody> </table>	Bits	Description	00b	No delay.	01b	1/3 maximum additional delay	10b	2/3 maximum additional delay	11b	Maximum additional delay
Bits	Description										
00b	No delay.										
01b	1/3 maximum additional delay										
10b	2/3 maximum additional delay										
11b	Maximum additional delay										
12:0	Reserved.										

**D18F2x9C\_x0D0F\_1C00\_dct[0] Clock Transmit PreDriver Calibration**

Cold reset: xxxx\_xxxxh.

**Table 156:** Broadcast write index address for D18F2x9C\_x0D0F\_1C00\_dct[0]

D18F2x98_dct[0][31:16]	D18F2x98_dct[0][15:0]
	1C00h
0D0Fh	D18F2x9C_x0D0F_0[F,8:0]0[B,7,3]_dct[0]

Bits	Description
31:14	Reserved.
13:8	<b>POdtD: pmos ODT calibration data.</b> Write-only.
7:6	Reserved.
5:0	<b>NOdtD: nmos ODT calibration data.</b> Write-only.

**D18F2x9C\_x0D0F\_2[2:0]02\_dct[0] Clock Transmit PreDriver Calibration**

Cold reset: xxxx\_xxxxh. BIOS: See 2.9.7.4.4.

**Table 157:** Index Mapping for D18F2x9C\_x0D0F\_2[2:0]02\_dct[0]

D18F2x98_dct[0][31:0]	Function
0D0F_2002h	Clock 0 Pad Group 0
0D0F_2102h	Clock 1 Pad Group 0
0D0F_2202h	Reserved

Bits	Description
31:16	Reserved.
15	<b>ValidTxAndPre: predriver calibration code valid.</b> Read-write; cleared-by-hardware. 1=Predriver calibration codes are copied from this register into the associated transmit pad. Hardware clears this field after the copy is complete.
14:12	Reserved.
11:6	<b>TxPreP: PMOS predriver calibration code.</b> Read-write. This field specifies the rising edge slew rate of the transmit pad. See: <a href="#">D18F2x9C_x0D0F_0[F,8:0]02_dct[0][TxPreP]</a> . After updating this value, BIOS must program ValidTxAndPre=1 for the change to take effect.
5:0	<b>TxPreN: NMOS predriver calibration code.</b> Read-write. This field specifies the falling edge slew rate of the transmit pad. See: <a href="#">D18F2x9C_x0D0F_0[F,8:0]02_dct[0][TxPreN]</a> . After updating this value, BIOS must program ValidTxAndPre=1 for the change to take effect.

### **D18F2x9C\_x0D0F\_[C,8,2][2:0]1E\_dct[0]\_mp[1:0] Phy Control**

Cold reset: 0000\_5020h.

Table 158: Index address mapping for [D18F2x9C\\_x0D0F\\_\[C,8,2\]\[2:0\]1E\\_dct\[0\]\\_mp\[1:0\]](#)

<a href="#">D18F2x98_dct[0][31:0]</a>	Function
0D0F_201Eh	Clock 0
0D0F_211Eh	Clock 1
0D0F_221Eh	Reserved
0D0F_801Eh	Cmd/Addr 0
0D0F_811Eh	Cmd/Addr 1
0D0F_821Eh	Reserved
0D0F_C01Eh	Address
0D0F_C11Eh	Reserved
0D0F_C21Eh	Reserved

Bits	Description
31:8	Reserved.
7	Reserved.
6:0	Reserved.

### **D18F2x9C\_x0D0F\_[C,8,2][2:0]1F\_dct[0] Receiver Configuration**

Cold reset: 0000\_2000h.

Table 159: Index address mapping for [D18F2x9C\\_x0D0F\\_\[C,8,2\]\[2:0\]1F\\_dct\[0\]](#)

<a href="#">D18F2x98_dct[0][31:0]</a>	Function
0D0F_201Fh	Clock 0
0D0F_211Fh	Clock 1
0D0F_221Fh	Reserved
0D0F_801Fh	Cmd/Addr 0
0D0F_811Fh	Cmd/Addr 1

Table 159: Index address mapping for D18F2x9C\_x0D0F\_[C,8,2][2:0]1F\_dct[0]

0D0F_821Fh	Reserved
0D0F_C01Fh	Address
0D0F_C11Fh	Reserved
0D0F_C21Fh	Reserved

Bits	Description										
31:5	Reserved.										
4:3	<p><b>RxVioLvl: receiver voltage level.</b> Read-write. BIOS: See 2.9.7.4.1. Specifies the VDDIO voltage level.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>1.5 V</td> </tr> <tr> <td>01b</td> <td>1.35 V</td> </tr> <tr> <td>10b</td> <td>1.25 V</td> </tr> <tr> <td>11b</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Description	00b	1.5 V	01b	1.35 V	10b	1.25 V	11b	Reserved
Bits	Description										
00b	1.5 V										
01b	1.35 V										
10b	1.25 V										
11b	Reserved										
2:0	Reserved.										

**D18F2x9C\_x0D0F\_[C,8,2][2:0]20\_dct[0]\_mp[1:0] DLL Delay and Configuration**

Cold reset: 0000\_8000h. Updated by hardware during 2.9.7.4.3 [Phy Fence Programming].

Table 160: Index Addresses for D18F2x9C\_x0D0F\_[C,8,2][2:0]20\_dct[0]\_mp[1:0]

D18F2x98_dct[0][31:0]	Function
0D0F_2020h	Clock 0
0D0F_2120h	Clock 1
0D0F_2220h	Reserved
0D0F_8020h	Cmd/Addr 0
0D0F_8120h	Cmd/Addr 1
0D0F_8220h	Reserved
0D0F_C020h	Address
0D0F_C120h	Reserved
0D0F_C220h	Reserved

Bits	Description
31:16	Reserved.
15	Reserved.
14	Reserved.
13:8	Reserved.
7	<b>FenceBit: fence bit.</b> Read-write.
6	<b>Fence2Bit: fence2 bit.</b> Read-write.
5:0	Reserved.

**D18F2x9C\_x0D0F\_2[F,2:0]30\_dct[0] Clock DLL Configuration and Power Down**

Cold reset: 0000\_0001h.

**Table 161: Index Addresses for D18F2x9C\_x0D0F\_2[F,2:0]30\_dct[0]**

D18F2x98_dct[0][31:0]	Function
0D0F_2030h	Clock 0
0D0F_2130h	Clock 1
0D0F_2230h	Reserved
0D0F_2F30h	Clock 1-0

Bits	Description
31:16	Reserved.
15:5	Reserved.
4	<b>PwrDn: power down.</b> Read-write. 1=Turn off DLL circuitry.
3:0	Reserved.

**D18F2x9C\_x0D0F\_4006\_dct[0] DRAM Phy MemVref Observation Configuration**

Cold reset: 0000\_0002h.

Bits	Description
31:16	Reserved.
15:9	Reserved.
8:3	Reserved.
2:0	<b>VrefSel: vref selection.</b> Read-write.
	<u>Bits</u> <u>Description</u>
	000b                      Connections to the bump and internal VREF bus are driven tri-state.
	001b                      Internal Rx. VrefDAC is sent to the DQ receivers; pad connection is tri-state.
	010b                      External Rx. External Vref from the pad is sent to the DQ receivers.
	011b                      Internal Tx. VrefDAC is sent out on the pad; internal connection is tri-state.
	110b-100b                Reserved.
	111b                      Internal Rx & Tx. VrefDAC is sent to all the DQ receivers and pad.

**D18F2x9C\_x0D0F\_4007\_dct[0] DRAM Phy MemVref Configuration**

Cold reset: 0000\_0002h.

Bits	Description
31:16	Reserved.
15:9	Reserved.

8:3	<b>VrefDAC: vref DAC.</b> Read-write. Controls the relative offset for the internal generated Vref from nominal VDDIO/2. <u>Bits</u> 1Eh-00h 20h-1Fh 3Eh-21h 3Fh	<u>Description</u> <VrefDAC>% Reserved 20h - <VrefDAC>% Reserved
2:0	<b>VrefSel: vref selection.</b> Read-write. <u>Bits</u> 000b 001b 010b 011b 110b-100b 111b	<u>Description</u> Reseved. Internal Rx. VrefDAC is sent to the DQ receivers; pad connection is tri-state. External Rx. External Vref from the pad is sent to the DQ receivers. Internal Tx. VrefDAC is sent out on the pad; internal connection is tri-state. Reserved. Internal Rx & Tx. VrefDAC is sent to all the DQ receivers and pad.

**D18F2x9C\_x0D0F\_4009\_dct[0] Phy Cmp Configuration**

Cold reset: 0000\_2000h.

Bits	Description	
31:16	Reserved.	
15:14	<b>CmpVioLvl: cmp voltage level.</b> Read-write. BIOS: See 2.9.7.4.1. This field specifies the VDDIO voltage level. <u>Bits</u> 00b 01b 10b 11b	<u>Description</u> 1.5 V 1.35 V 1.25 V Reserved
13:4	Reserved.	
3:2	<b>ComparatorAdjust: comparator adjust.</b> Read-write. BIOS: <a href="#">D18F2x9C_x0D0F_0[F,8:0]1F_dct[0]_mp[1:0][RxVioLvl]</a> . This field specifies the adjustment signals for the comparator differential amplifier.	
1:0	Reserved.	

**D18F2x9C\_x0D0F\_[C,8][1:0]02\_dct[0] Transmit PreDriver Calibration**

Cold reset: xxxx\_xxxxh. BIOS: See 2.9.7.4.4.

Table 162: [Index Mapping](#) for [D18F2x9C\\_x0D0F\\_\[C,8\]\[1:0\]02\\_dct\[0\]](#)

D18F2x98_dct[0][31:0]	Function
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	<b>ValidTxAndPre: predriver calibration code valid.</b> Read-write; cleared-by-hardware. 1=Predriver calibration codes are copied from this register and <a href="#">D18F2x9C_x0D0F_[C,8][1:0][12,0E,0A,06]_dct[0]</a> into the associated transmit pad. Hardware clears this field after the copy is complete.
14:12	Reserved.
11:6	<b>TxPreP: PMOS predriver calibration code.</b> Read-write. This field specifies the rising edge slew rate of the transmit pad. See: <a href="#">D18F2x9C_x0D0F_0[F,8:0]02_dct[0][TxPreP]</a> . After updating this value, BIOS must program ValidTxAndPre=1 for the change to take effect.
5:0	<b>TxPreN: NMOS predriver calibration code.</b> Read-write. This field specifies the falling edge slew rate of the transmit pad. See: <a href="#">D18F2x9C_x0D0F_0[F,8:0]02_dct[0][TxPreN]</a> . After updating this value, BIOS must program ValidTxAndPre=1 for the change to take effect.

**D18F2x9C\_x0D0F\_[C,8][1:0][12,0E,0A,06]\_dct[0] Transmit PreDriver Calibration 2**

Cold reset: xxxx\_xxxxh. BIOS: See 2.9.7.4.4.

Table 163: Index Mapping for [D18F2x9C\\_x0D0F\\_\[C,8\]\[1:0\]\[12,0E,0A,06\]\\_dct\[0\]](#)

<a href="#">D18F2x98_dct[0][31:0]</a>	Function
0D0F_8006h	Cmd/Addr 0 Pad Group 1
0D0F_800Ah	Cmd/Addr 0 Pad Group 2
0D0F_8106h	Cmd/Addr 1 Pad Group 1
0D0F_810Ah	Cmd/Addr 1 Pad Group 2
0D0F_C006h	Address Pad Group 1
0D0F_C00Ah	Address Pad Group 2
0D0F_C00Eh	Address Pad Group 3
0D0F_C012h	Address Pad Group 4

Bits	Description
31:12	Reserved.
11:6	<b>TxPreP: PMOS predriver calibration code.</b> Read-write. This field specifies the rising edge slew rate of the transmit pad. See: <a href="#">D18F2x9C_x0D0F_0[F,8:0]02_dct[0][TxPreP]</a> . After updating this value, BIOS must program <a href="#">D18F2x9C_x0D0F_[C,8][1:0]02_dct[0][ValidTxAndPre]=1</a> for the change to take effect.
5:0	<b>TxPreN: NMOS predriver calibration code.</b> Read-write. This field specifies the falling edge slew rate of the transmit pad. See: <a href="#">D18F2x9C_x0D0F_0[F,8:0]02_dct[0][TxPreN]</a> . After updating this value, BIOS must program <a href="#">D18F2x9C_x0D0F_[C,8][1:0]02_dct[0][ValidTxAndPre]=1</a> for the change to take effect.

**D18F2x9C\_x0D0F\_812F\_dct[0] Tristate Configuration**

Reset: 0000\_00A0h. BIOS: See 2.9.7.10.

Bits	Description
31:8	Reserved.
7	<b>Add16Tri: MEMADD[16] tri-state.</b> Read-write. Specifies tri-state control for the memory address[16] signal. 1=Signal is tri-stated. 0=Signal is not tri-stated.

6	Reserved.
5	<b>Add17Tri: MEMADD[17] tri-state.</b> Read-write. 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	<b>PARTri: MEMPAR tri-state.</b> Read-write. Specifies tri-state control for the memory parity signal. 1=Signal is tri-stated. 0=Signal is not tri-stated.

### **D18F2x9C\_x0D0F\_[C,8][F:0]30\_dct[0] Cmd/Addr DLL Configuration and Power Down**

Cold reset: 0000\_0001h. BIOS: See 2.9.7.4.4.

Table 164: Index Mapping for D18F2x9C\_x0D0F\_[C,8][F:0]30\_dct[0]

D18F2x98_dct[0][31:0]	Function
0D0F_8030h	Cmd/Addr 0
0D0F_8130h	Cmd/Addr 1
0D0F_8F30h	Cmd/Addr 0 and Cmd/Addr 1
0D0F_C030h	Address 2

Bits	Description
31:16	Reserved.
15:5	Reserved.
4	<b>PwrDn: power down.</b> Read-write. 1=Turn off DLL circuitry.
3:0	Reserved.

### **D18F2x9C\_x0D0F\_C021\_dct[0]\_mp[1:0] DLL Delay and Configuration**

Cold reset: 0000\_0000h. Updated by hardware during 2.9.7.4.3 [Phy Fence Programming].

Bits	Description
31:16	Reserved.
15:8	Reserved.
7	<b>FenceBit: fence bit.</b> Read-write.
6	<b>Fence2Bit: fence2 bit.</b> Read-write.
5:0	Reserved.

### **D18F2x9C\_x0D0F\_E000\_dct[0]\_mp[1:0] Phy Master Configuration**

Cold reset: 0000\_0002h.

Bits	Description
31:16	Reserved.
15:14	<b>Rate[4:3].</b> See: Rate[2:0].



13	<b>POdtOff: processor ODT off.</b> Read-write. BIOS: See <a href="#">D18F2x9C_x0D0F_0[F,8:0]04_dct[0]_mp[1:0][POdtOff]</a> . 1=Phy disables receiver pad termination. 0=Phy enables receiver pad termination.
12:4	Reserved.
3	<b>FreqValid: Frequency valid.</b> Read-write. 1=The memory frequency specified with the Rate field is valid. software must set this bit when programming Rate for the setting to take effect. Regardless of this bit, a new rate does not take effect unless the target register context is of the current memory P-state, or until the memory P-state is changed to the target Rate context.
2:0	<b>Rate[2:0].</b> Read-write. Rate[4:0] = {Rate[4:3], Rate[2:0]}. Rate[4:0] is the 5-bit DDR data rate that the phy uses to configure the PLL. See <a href="#">Table 125 [Valid Values for Memory Clock Frequency Value Definition]</a> . Software must set FreqValid=1 when programming this field. If memory P-states are enabled then software must program the M1 context. It is not necessary for software to program the M0 context when using <a href="#">D18F2x94_dct[0][MemClkFreqVal]</a> .

#### **D18F2x9C\_x0D0F\_E006\_dct[0] Phy PLL Lock Time**

Cold reset: 0000\_0190h.

Bits	Description
31:16	Reserved.
15:0	<b>PIILockTime: pll lock time.</b> Read-write. Specifies the number of 5ns periods the phy waits for PLLs to lock during a frequency change. See <a href="#">2.9.7.4.2 [DRAM Channel Frequency Change]</a> .

#### **D18F2x9C\_x0D0F\_E007\_dct[0] Phy DLL Lock Time**

Cold reset: 0000\_0000h.

Bits	Description
31:16	Reserved.
15:0	<b>DILLockTime: dll lock time.</b> Read-write. Specifies the number of RefClk cycles the phy waits for DLLs to lock during a frequency change. See <a href="#">2.9.7.4.2 [DRAM Channel Frequency Change]</a> . The function of this CSR is replaced with DLL Gain Sequencer. This timer runs in parallel with the gain sequencer and the status of DLL is marked as locked when the gain sequencer has completed its sequence and the DLL Lock Time counter has also expired.

#### **D18F2x9C\_x0D00\_E008\_dct[0] Phy Master Configuration**

Cold reset: 0000\_0013h.

Bits	Description
31:13	Reserved.
12	<b>PhyPS: Phy memory P-state.</b> Read-only; updated-by-hardware. Indicates the current memory P-state for the phy. 0=M0. 1=M1.
11:5	Reserved.
4:0	<b>FenceValue: fence value.</b> Read-only; updated-by-hardware. Indicates the fence value used to create the fence bit in the DLL delay registers. See <a href="#">2.9.7.4.3 [Phy Fence Programming]</a> .

**D18F2x9C\_x0D04\_E008\_dct[0] Phy Master Configuration**

Cold reset: 0000\_0013h.

Bits	Description
31:9	Reserved.
8	<b>PStateToAccess: P-state to access.</b> Read-write. Specifies the memory P-state context for context sensitive DDR phy CSR accesses using <a href="#">D18F2x98_dct[0]</a> when DctOffset[29:20] != 0D0h. 0=M0, 1=M1. The processor updates this field during a memory P-state change with the target state. See <a href="#">D18F2x94_dct[0][DphyMemPsSelEn]</a> .
7:0	Reserved.

**D18F2x9C\_x0D0F\_E00A\_dct[0] Phy Dynamic Power Mode**

Cold reset: 0000\_0000h.

Bits	Description
31:5	Reserved.
4	Reserved.
3:0	Reserved.

**D18F2x9C\_x0D0F\_E013\_dct[0] Phy PLL Regulator Wait Time**

Cold reset: 0000\_0118h.

Bits	Description
31:16	Reserved.
15:0	<b>PllRegWaitTime: PLL regulator wait time.</b> Read-write. BIOS: <a href="#">2.9.7.4</a> . 1=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.

**D18F2x9C\_x0D0F\_E019\_dct[0] Fence2**Cold reset: 0000\_0000h. BIOS: [2.9.7.4.3](#).

Bits	Description
31:15	Reserved.
14	<b>Fence2EnableRxDll: phy fence2 enable receive DLL.</b> Read-write. 1=Enable the use of Fence2ThresholdRxDll for DQS receiver enable second fence threshold. 0=Disable second fence. See <a href="#">D18F2x9C_x0000_000C_dct[0][FenceThresholdRxDll]</a> for first fence.
13:10	<b>Fence2ThresholdRxDll: phy fence2 threshold DQS receiver enable.</b> Read-write. If Fence2EnableRxDll=1, this field specifies the fence delay threshold value used for DQS receiver enable. See Fence2ThresholdTxPad.
9:5	Reserved.

4	<b>Fence2EnableTxPad: fence2 enable transmit pad.</b> Read-write. 1=Enable the use of Fence2ThresholdTxPad for transmit pad second fence threshold. 0=Disable second fence. See <a href="#">D18F2x9C_x0000_000C_dct[0][FenceThresholdTxPad]</a> for first fence.										
3:0	<b>Fence2ThresholdTxPad: phy fence2 threshold transmit pad.</b> Read-write. If Fence2EnableTxPad=1, this field specifies the second fence delay threshold value used for command, address, write data and write DQS. <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>16/64 MEMCLK delay</td> </tr> <tr> <td>1h</td> <td>17/64 MEMCLK delay</td> </tr> <tr> <td>Eh-2h</td> <td>{1,&lt;Fence2ThresholdTxPad&gt;}/64 MEMCLK delay</td> </tr> <tr> <td>Fh</td> <td>31/64 MEMCLK delay</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	0h	16/64 MEMCLK delay	1h	17/64 MEMCLK delay	Eh-2h	{1,<Fence2ThresholdTxPad>}/64 MEMCLK delay	Fh	31/64 MEMCLK delay
<u>Bits</u>	<u>Definition</u>										
0h	16/64 MEMCLK delay										
1h	17/64 MEMCLK delay										
Eh-2h	{1,<Fence2ThresholdTxPad>}/64 MEMCLK delay										
Fh	31/64 MEMCLK delay										

### **D18F2x9C\_x0D0F\_E01A\_dct[0] M1 Fence Value**

Cold reset: 0000\_0000h. BIOS: [2.9.7.4.3](#). This register is used by BIOS to temporarily store the trained fence values for the M1 memory P-state, and has no effect on the operation of the hardware. See [2.9.7.4.3](#).

Bits	Description
31:16	Reserved.
15	Reserved.
14:10	<b>TxDll: Transmit dll fence value.</b> Read-write. BIOS saves the trained TxDll to this field until such time as it is able to write the value to NVRAM.
9:5	<b>RxDll: Receiver dll fence value.</b> Read-write. BIOS saves the trained RxDll to this field until such time as it is able to write the value to NVRAM.
4:0	<b>TxPad: Transmit pad fence value.</b> Read-write. BIOS saves the trained TxPad to this field until such time as it is able to write the value to NVRAM.

**D18F2xA4 DRAM Controller Temperature Throttle**

See 2.9.3 [DCT Configuration Registers].

See 2.9.12 [DRAM On DIMM Thermal Management and Power Capping].

Bits	Description																																		
31:24	Reserved.																																		
23:20	<p><b>BwCapCmdThrottleMode: bandwidth capping command throttle mode.</b> Read-write. Reset: 0. Specifies the command throttle mode when BwCapEn=1. The DCT throttles commands over a rolling window of 100 clock cycles, maintaining the average throttling as specified by this field.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr><td>0000b</td><td>Command throttling is disabled</td></tr> <tr><td>0001b</td><td>Throttle commands by 30%</td></tr> <tr><td>0010b</td><td>Throttle commands by 40%</td></tr> <tr><td>0011b</td><td>Throttle commands by 50%</td></tr> <tr><td>0100b</td><td>Throttle commands by 55%</td></tr> <tr><td>0101b</td><td>Throttle commands by 60%</td></tr> <tr><td>0110b</td><td>Throttle commands by 65%</td></tr> <tr><td>0111b</td><td>Throttle commands by 70%</td></tr> <tr><td>1000b</td><td>Throttle commands by 75%</td></tr> <tr><td>1001b</td><td>Throttle commands by 80%</td></tr> <tr><td>1010b</td><td>Throttle commands by 85%</td></tr> <tr><td>1011b</td><td>Throttle commands by 90%</td></tr> <tr><td>1100b</td><td>Throttle commands by 95%</td></tr> <tr><td>1101b</td><td>Reserved</td></tr> <tr><td>1110b</td><td>Throttle commands as specified by CmdThrottleMode</td></tr> <tr><td>1111b</td><td>Reserved</td></tr> </tbody> </table> <p>Throttling should not be enabled until after DRAM initialization (<a href="#">D18F2x110[DramEnable]=1</a>) and training (see 2.9.7.9 [DRAM Training]) are complete.</p>	Bits	Description	0000b	Command throttling is disabled	0001b	Throttle commands by 30%	0010b	Throttle commands by 40%	0011b	Throttle commands by 50%	0100b	Throttle commands by 55%	0101b	Throttle commands by 60%	0110b	Throttle commands by 65%	0111b	Throttle commands by 70%	1000b	Throttle commands by 75%	1001b	Throttle commands by 80%	1010b	Throttle commands by 85%	1011b	Throttle commands by 90%	1100b	Throttle commands by 95%	1101b	Reserved	1110b	Throttle commands as specified by CmdThrottleMode	1111b	Reserved
Bits	Description																																		
0000b	Command throttling is disabled																																		
0001b	Throttle commands by 30%																																		
0010b	Throttle commands by 40%																																		
0011b	Throttle commands by 50%																																		
0100b	Throttle commands by 55%																																		
0101b	Throttle commands by 60%																																		
0110b	Throttle commands by 65%																																		
0111b	Throttle commands by 70%																																		
1000b	Throttle commands by 75%																																		
1001b	Throttle commands by 80%																																		
1010b	Throttle commands by 85%																																		
1011b	Throttle commands by 90%																																		
1100b	Throttle commands by 95%																																		
1101b	Reserved																																		
1110b	Throttle commands as specified by CmdThrottleMode																																		
1111b	Reserved																																		
19:15	Reserved.																																		
14:12	<p><b>CmdThrottleMode: command throttle mode.</b> Read-write. Reset: 0. BIOS: See 2.9.7.7 [DCT Training Specific Configuration]. Specifies the command throttle mode when ODTSEn=1 and the EVENT_L pin is asserted. The DCT throttles commands over a rolling window of 100 clock cycles, maintaining the average throttling as specified by this field.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr><td>000b</td><td>Command throttling is disabled.</td></tr> <tr><td>001b</td><td>Throttle commands by 30%.</td></tr> <tr><td>010b</td><td>Throttle commands by 50%.</td></tr> <tr><td>011b</td><td>Throttle commands by 60%.</td></tr> <tr><td>100b</td><td>Throttle commands by 70%.</td></tr> <tr><td>101b</td><td>Throttle commands by 80%.</td></tr> <tr><td>110b</td><td>Throttle commands by 90%.</td></tr> <tr><td>111b</td><td>Place the DRAM devices in powerdown mode (see <a href="#">D18F2x94_dct[0][PowerDownMode]</a>) when EVENT_L is asserted. This mode is not valid if <a href="#">D18F2x94_dct[0][PowerDownEn]=0</a>.</td></tr> </tbody> </table> <p>Throttling should not be enabled until after DRAM initialization (<a href="#">D18F2x110[DramEnable]=1</a>) and training (See 2.9.7.9 [DRAM Training]) are complete. See also BwCapEn.</p>	Bits	Description	000b	Command throttling is disabled.	001b	Throttle commands by 30%.	010b	Throttle commands by 50%.	011b	Throttle commands by 60%.	100b	Throttle commands by 70%.	101b	Throttle commands by 80%.	110b	Throttle commands by 90%.	111b	Place the DRAM devices in powerdown mode (see <a href="#">D18F2x94_dct[0][PowerDownMode]</a> ) when EVENT_L is asserted. This mode is not valid if <a href="#">D18F2x94_dct[0][PowerDownEn]=0</a> .																
Bits	Description																																		
000b	Command throttling is disabled.																																		
001b	Throttle commands by 30%.																																		
010b	Throttle commands by 50%.																																		
011b	Throttle commands by 60%.																																		
100b	Throttle commands by 70%.																																		
101b	Throttle commands by 80%.																																		
110b	Throttle commands by 90%.																																		
111b	Place the DRAM devices in powerdown mode (see <a href="#">D18F2x94_dct[0][PowerDownMode]</a> ) when EVENT_L is asserted. This mode is not valid if <a href="#">D18F2x94_dct[0][PowerDownEn]=0</a> .																																		

11	<b>BwCapEn: bandwidth capping enable.</b> Read-write. Reset: 0. 1=The memory command throttle mode specified by BwCapCmdThrottleMode is applied. This bit is used by software to enable command throttling independent of the state of the EVENT_L pin. If this bit is set, ODTSEn=1, and the EVENT_L pin is asserted, the larger of the two throttle percentages specified by CmdThrottleMode and BwCapCmdThrottleMode is used.
10:9	Reserved.
8	<b>ODTSEn: on DIMM temperature sensor enable.</b> Read-write. Reset: 0. BIOS: See 2.9.7.7 [DCT Training Specific Configuration]. Enables the monitoring of the EVENT_L pin and indicates whether the on DIMM temperature sensors of the DIMMs on a channel are enabled. 0=Disabled. 1=Enabled. While the EVENT_L pin is asserted, the controller (a) doubles the refresh rate (if Tref=7.8 us), and (b) throttles the address bus utilization as specified by CmdThrottleMode[2:0].
7:0	Reserved.

### D18F2xA8\_dct[0] DRAM Controller Miscellaneous 2

See 2.9.3 [DCT Configuration Registers].

Bits	Description
31	<b>PerRankTimingEn: per rank timing enable.</b> Read-write. Reset: 0. BIOS: 1. Specifies the mapping between chip selects and a set of programmable timing delays. 1=Each chip select is controlled by a set of timing delays. A maximum of 4 chip selects are supported per channel. 0=Each chip select pair is controlled by a set timing delays.
30	Reserved.
29	Reserved.
28	<b>FastSelfRefEntryDis: fast self refresh entry disable.</b> Read-write; Same-for-all. Reset: 1. BIOS: ~D18F2x1B4[FlushWrOnS3StpGnt]. 1=DCT pushes outstanding transactions to memory prior to entering self refresh. 0=DCT enters self refresh immediately unless instructed to push outstanding transactions to memory by D18F2x11C[FlushWrOnStpGnt] or D18F2x1B4[FlushWrOnS3StpGnt].
27:26	Reserved.
25:24	Reserved.
23	Reserved.
22	<b>PrtlChPDEnhEn: partial channel power down enh enable.</b> Read-write. Reset: 0. BIOS: 0. Selects the channel idle hysteresis for fast exit/slow exit mode changes when (D18F2x94_dct[0][PowerDownMode] & D18F2x84_dct[0][PchgPDMoSel]). 1=Hysteresis specified by D18F2x244_dct[0][PrtlChPDDynDly]. 0=256 clock hysteresis.
21	<b>AggrPDEn: aggressive power down enable.</b> Read-write. Reset: 0. BIOS: 1. 1=The DCT places the DRAM devices in precharge power down mode when pages are closed as specified by D18F2x248_dct[0]_mp[1:0][AggrPDDelay]. 0=The DCT places the DRAM devices in precharge power down mode when pages are closed as specified by D18F2x90_dct[0][DynPageCloseEn].
20	<b>BankSwap: swap bank address.</b> Read-write. Reset: 0. BIOS: See 2.9.7.7. 1=Swap the DRAM bank address bits. The normalized address bits used for bank address are swapped with lower order address bits. If D18F2x114[DctSelBankSwap]=1 then normalized address bits 10:8 are swapped else normalized address bits 11:9 are swapped. See D18F2x80_dct[0]. This swap happens before D18F2x94_dct[0][BankSwizzleMode] is applied.

19	Reserved.										
18	Reserved.										
17:16	<p><b>MemPhyPllPdMode: memory phy PLL powerdown mode.</b> Read-write. Reset: 00b. BIOS: 10b. These bits control how the memory PLL powers down during self-refresh. If self-refresh is requested for an NB p-state change, then the memory PLL is not powered down. Memory phy PLL powerdown can only be enabled if NB PLL power down is enabled. See <a href="#">D18F5x128[NbPllPwrDwnRegEn]</a>.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>PLL powerdown is disabled</td> </tr> <tr> <td>01b</td> <td>PLL VCO powerdown during SR</td> </tr> <tr> <td>10b</td> <td>PLL regulator powerdown during SR</td> </tr> <tr> <td>11b</td> <td>PLL VCO and regulator powerdown during SR</td> </tr> </tbody> </table>	Bits	Description	00b	PLL powerdown is disabled	01b	PLL VCO powerdown during SR	10b	PLL regulator powerdown during SR	11b	PLL VCO and regulator powerdown during SR
Bits	Description										
00b	PLL powerdown is disabled										
01b	PLL VCO powerdown during SR										
10b	PLL regulator powerdown during SR										
11b	PLL VCO and regulator powerdown during SR										
15:8	Reserved.										
7:6	Reserved.										
5	Reserved.										
4	Reserved.										
3:2	Reserved.										
1:0	Reserved.										

**D18F2xAC DRAM Controller Temperature Status**

Cold reset: 0000\_0000h.

Bits	Description
31:8	Reserved.
7:4	Reserved.
3:2	Reserved.
1	Reserved.
0	<p><b>MemTempHot0: Memory temperature hot, DCT0.</b> Read; Write-1-to-clear. 1=The EVENT_L pin was asserted indicating the memory temperature exceeded the normal operating limit; the DCT may be throttling the interface to aid in cooling. See <a href="#">D18F2xA4</a>.</p>

**D18F2xF8 P-state Power Information 1**

Read-only.

Bits	Description
31:24	<p><b>PwrValue3: P3 power value.</b> See PwrValue0. Value: Product-specific.</p>
23:16	<p><b>PwrValue2: P2 power value.</b> See PwrValue0. Value: Product-specific.</p>

15:8	<b>PwrValue1: P1 power value.</b> See PwrValue0. Value: Product-specific.										
7:0	<b>PwrValue0: P0 power value.</b> PwrValue and PwrDiv together specify the expected power draw of a single core in P0 and 1/NumCores of the Northbridge in the NB P-state as specified by <a href="#">MSRC001_00[6B:64][NbPstate]</a> . NumCores is defined to be the number of cores per node at cold reset. Value: Product-specific. <table border="1"> <thead> <tr> <th><u>PwrDiv</u></th> <th><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>PwrValue / 1 W, Range: 0 to 255 W</td> </tr> <tr> <td>01b</td> <td>PwrValue / 10 W, Range: 0 to 25.5 W</td> </tr> <tr> <td>10b</td> <td>PwrValue / 100 W, Range: 0 to 2.55 W</td> </tr> <tr> <td>11b</td> <td>Reserved</td> </tr> </tbody> </table>	<u>PwrDiv</u>	<u>Description</u>	00b	PwrValue / 1 W, Range: 0 to 255 W	01b	PwrValue / 10 W, Range: 0 to 25.5 W	10b	PwrValue / 100 W, Range: 0 to 2.55 W	11b	Reserved
<u>PwrDiv</u>	<u>Description</u>										
00b	PwrValue / 1 W, Range: 0 to 255 W										
01b	PwrValue / 10 W, Range: 0 to 25.5 W										
10b	PwrValue / 100 W, Range: 0 to 2.55 W										
11b	Reserved										

### D18F2xFC P-state Power Information 2

Read-only.

Bits	Description
31:24	Reserved.
23:22	<b>PwrDiv7: P7 power divisor.</b> See <a href="#">D18F2xF8[PwrValue0]</a> . Value: Product-specific.
21:20	<b>PwrDiv6: P6 power divisor.</b> See <a href="#">D18F2xF8[PwrValue0]</a> . Value: Product-specific.
19:18	<b>PwrDiv5: P5 power divisor.</b> See <a href="#">D18F2xF8[PwrValue0]</a> . Value: Product-specific.
17:16	<b>PwrDiv4: P4 power divisor.</b> See <a href="#">D18F2xF8[PwrValue0]</a> . Value: Product-specific.
15:14	<b>PwrDiv3: P3 power divisor.</b> See <a href="#">D18F2xF8[PwrValue0]</a> . Value: Product-specific.
13:12	<b>PwrDiv2: P2 power divisor.</b> See <a href="#">D18F2xF8[PwrValue0]</a> . Value: Product-specific.
11:10	<b>PwrDiv1: P1 power divisor.</b> See <a href="#">D18F2xF8[PwrValue0]</a> . Value: Product-specific.
9:8	<b>PwrDiv0: P0 power divisor.</b> See <a href="#">D18F2xF8[PwrValue0]</a> . Value: Product-specific.
7:0	<b>PwrValue4: P4 power value.</b> See <a href="#">D18F2xF8[PwrValue0]</a> . Value: Product-specific.

### D18F2x104 P-state Power Information 3

Read-only.

Bits	Description
31:24	Reserved.
23:16	<b>PwrValue7: P7 power value.</b> See <a href="#">D18F2xF8[PwrValue0]</a> . Value: Product-specific.
15:8	<b>PwrValue6: P6 power value.</b> See <a href="#">D18F2xF8[PwrValue0]</a> . Value: Product-specific.
7:0	<b>PwrValue5: P5 power value.</b> See <a href="#">D18F2xF8[PwrValue0]</a> . Value: Product-specific.

### D18F2x110 DRAM Controller Select Low

Reset: 0000\_0000h.

Bits	Description
31:11	Reserved.
10	<b>MemCleared: memory cleared.</b> Read-only; updated-by-hardware. 1=Memory has been cleared since the last warm reset. This bit is set by MemClrInit. See MemClrInit.

9	<b>MemClrBusy: memory clear busy.</b> Read-only; updated-by-hardware. 1=The 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	<b>DramEnable: DRAM enabled.</b> Read-only. 1=All of the used DCTs are initialized (see 2.9.7.8 [DRAM Device and Controller Initialization]) or have exited from self refresh (D18F2x90_dct[0][ExitSelfRef] transitions from 1 to 0).
7:6	Reserved.
5	<b>DctDatIntLv: DRAM controller data interleave enable.</b> IF (D18F2x118[LockDramCfg]) THEN Read-only. ELSE Read-write. ENDIF. BIOS: D18F3x44[DramEccEn]. 1=DRAM data bits from every two consecutive 64-bit DRAM lines are interleaved in the ECC calculation such that a dead bit of a DRAM device is correctable.
4	Reserved.
3	<b>MemClrInit: memory clear initialization.</b> IF (D18F2x118[LockDramCfg]) THEN Read-only. ELSE Read-write. ENDIF. 1=The node writes 0's to all locations of system memory attached to the node and sets the MemCleared bit. The status of the memory clear operation can be determined by reading the MemClrBusy and MemCleared bits. This command is ignored if MemClrBusy=1 when the command is received. DramEnable must be set before setting MemClrInit. The memory prefetcher must be disabled by setting D18F2x11C[PrefIoDis] and D18F2x11C[PrefCpuDis] before memory clear initialization and then can be re-enabled when MemCleared=1. Memory hole remapping must be disabled before setting MemClrInit. See D18F1x2[1,0][8,0][LgcyMmioHoleEn]. Software may then reenable memory hole remapping when the memory clear is complete.
2:0	Reserved.

### D18F2x114 DRAM Controller Select High

IF (D18F2x118[LockDramCfg]) THEN Read-only. ELSE Read-write. ENDIF. Reset: 0000\_0000h.

Bits	Description
31:10	Reserved.
9	<b>DctSelBankSwap: select DRAM bank swap address.</b> BIOS: 1. See D18F2xA8_dct[0][BankSwap].
8:0	Reserved.



**D18F2x118 Memory Controller Configuration Low**

Fields in this register (bits[17:0]) indicate priority of request types. Variable priority requests enter the memory controller 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 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.

Bits	Description																																				
31:28	<b>MctVarPriCntLmt: variable priority time limit.</b> Read-write. Reset: 0000b. BIOS: 0001b. <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <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> </tbody> </table>	Bits	Description	Bits	Description	0000b	80ns	1000b	720ns	0001b	160ns	1001b	800ns	0010b	240ns	1010b	880ns	0011b	320ns	1011b	960ns	0100b	400ns	1100b	1040ns	0101b	480ns	1101b	1120ns	0110b	560ns	1110b	1200ns	0111b	640ns	1111b	1280ns
Bits	Description	Bits	Description																																		
0000b	80ns	1000b	720ns																																		
0001b	160ns	1001b	800ns																																		
0010b	240ns	1010b	880ns																																		
0011b	320ns	1011b	960ns																																		
0100b	400ns	1100b	1040ns																																		
0101b	480ns	1101b	1120ns																																		
0110b	560ns	1110b	1200ns																																		
0111b	640ns	1111b	1280ns																																		
27	Reserved.																																				
26:24	<b>McqHiPriByPassMax: memory controller high priority bypass max.</b> Read-write. Reset: 100b. Specifies the number of times a medium- or low-priority DRAM request may be bypassed by high-priority DRAM requests.																																				
23	Reserved.																																				
22:20	<b>McqMedPriByPassMax: memory controller medium bypass low priority max.</b> Read-write. Reset: 100b. Specifies the number of times a low-priority DRAM request may be bypassed by medium-priority DRAM requests.																																				
19	<b>LockDramCfg.</b> Write-1-only. Reset: 0. BIOS: See 2.9.11 [DRAM CC6/PC6 Storage], 2.5.3.2.3.3 [Core C6 (CC6) State]. The following registers are read-only if LockDramCfg=1; otherwise the access type is specified by the register: <ul style="list-style-type: none"> <li>• D18F1xF0 [DRAM Hole Address]</li> <li>• D18F2x[5C:40]_dct[0] [DRAM CS Base Address]</li> <li>• D18F2x[6C:60]_dct[0] [DRAM CS Mask]</li> <li>• D18F2x80_dct[0] [DRAM Bank Address Mapping]</li> <li>• D18F2x110 [DRAM Controller Select Low]</li> <li>• D18F2x114 [DRAM Controller Select High]</li> <li>• D18F2x250_dct[0] [DRAM Loopback and Training Control]</li> <li>• D18F4x128[CoreStateSaveDestNode]</li> <li>• D18F1x[17C:140,7C:40] [DRAM Base/Limit]</li> <li>• D18F1x120 [DRAM Base System Address]</li> <li>• D18F1x124 [DRAM Limit System Address]</li> <li>• D18F2x118[CC6SaveEn]</li> </ul>																																				

18	<b>CC6SaveEn.</b> IF (D18F2x118[LockDramCfg]) THEN Read-only. ELSE Read-write. ENDIF. Reset: 0. 1=CC6 save area is enabled. See 2.5.3.2.7 [BIOS Requirements for Initialization], BIOS: (D18F4x118[PwrGateEnCstAct0]   D18F4x118[PwrGateEnCstAct1]   D18F4x11C[PwrGateEnCstAct2]).										
17:16	<b>MctPriScrub: scrubber priority.</b> Read-write. Reset: 00b. <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Medium</td> </tr> <tr> <td>01b</td> <td>Reserved</td> </tr> <tr> <td>10b</td> <td>High</td> </tr> <tr> <td>11b</td> <td>Variable</td> </tr> </tbody> </table>	Bits	Description	00b	Medium	01b	Reserved	10b	High	11b	Variable
Bits	Description										
00b	Medium										
01b	Reserved										
10b	High										
11b	Variable										
15:14	<b>MctPriTrace: trace-mode request priority.</b> See: MctPriCpuRd. Read-write. Reset: 10b. This must be set to high.										
13:12	<b>MctPriIsoc: display refresh read priority.</b> See: MctPriCpuRd. Read-write. Reset: 10b.										
11:10	<b>MctPriWr: default write priority.</b> See: MctPriCpuRd. Read-write. Reset: 01b.										
9:8	<b>MctPriDefault: default non-write priority.</b> See: MctPriCpuRd. Read-write. Reset: 00b.										
7:6	<b>MctPriIsocWr: IO write with the isoch bit set priority.</b> See: MctPriCpuRd. Read-write. Reset: 00b. This does not apply to isochronous traffic that is classified as display refresh.										
5:4	<b>MctPriIsocRd: IO read with the isoch bit set priority.</b> See: MctPriCpuRd. Read-write. Reset: 10b. This does not apply to isochronous traffic that is classified as display refresh.										
3:2	<b>MctPriCpuWr: CPU write priority.</b> See: MctPriCpuRd. Read-write. Reset: 01b.										
1:0	<b>MctPriCpuRd: CPU read priority.</b> Read-write. Reset: 00b. <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Medium</td> </tr> <tr> <td>01b</td> <td>Low</td> </tr> <tr> <td>10b</td> <td>High</td> </tr> <tr> <td>11b</td> <td>Variable</td> </tr> </tbody> </table>	Bits	Description	00b	Medium	01b	Low	10b	High	11b	Variable
Bits	Description										
00b	Medium										
01b	Low										
10b	High										
11b	Variable										

### D18F2x11C Memory Controller Configuration High

The two main functions of this register are to control write bursting and memory prefetching.

**Write bursting.** DctWrLimit and MctWrLimit specify how writes may be burst from the MCT into the DCT to improve DRAM efficiency. When the number of writes in the MCT reaches the value specified in MctWrLimit, then they are all burst to the DCTs at once. Prior to reaching the watermark, a limited number of writes can be passed to the DCTs (specified by DctWrLimit), tagged as low priority, for the DCTs to complete when otherwise idle. 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 MCQ until the threshold specified by MctWrLimit is reached.
- Once the threshold is reached, all writes in MCQ are converted to medium priority.
- Any write in MCQ that matches the address of a subsequent access is promoted to either medium priority or the priority of the subsequent access, whichever is higher.
- DctWrLimit only applies to low-priority writes.

**Memory prefetching.** The MCT 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 cachelines apart. The prefetcher supports strides of -4 to +4 cachelines, which can include alternating patterns (e.g. +1, +2, +1, +2), and can prefetch 1, 2, 3, 4, or 5 cachelines ahead,

depending on the confidence. In addition, a fixed stride mode (non-alternating) may be used for IO requests which often have fixed stride patterns. This mode bypasses the stride predictor such that CPU-access stride predictions are not adversely affected by IO streams.

The MCT tracks several stride patterns simultaneously. Each of these has a confidence level associated with it that varies as follows:

- Each time a request is received that matches the stride pattern, the confidence level increases by one.
- Each time a request is received within +/- 4 cachelines of the last requested cacheline in the pattern that does not match the pattern, then the confidence level decreases by one.
- When the confidence level reaches the saturation point specified by PrefConfSat, then it no-longer increments.

Each request that is not within +/- 4 cachelines of the last requested cacheline line of all the stride patterns tracked initiates a new stride pattern by displacing one of the existing least-recently-used stride patterns.

The memory prefetcher uses an adaptive prefetch scheme to adjust the prefetch distance based upon the buffer space available for prefetch request data. The adaptive scheme counts the total number of prefetch requests and the number of prefetch requests that cannot return data because of buffer availability. After every 16 prefetch requests, the prefetcher uses the following rules to adjust the prefetch distance:

- If the ratio of prefetch requests that cannot return data to total prefetch requests is greater than or equal to  $D18F2x1B0[AdapPrefMissRatio]$  then the prefetch distance is reduced by  $D18F2x1B0[AdapPrefNegativeStep]$ .
- If the ratio of prefetch requests that cannot return data to total prefetch requests is less than  $D18F2x1B0[AdapPrefMissRatio]$  then the prefetch distance is increased by  $D18F2x1B0[AdapPrefPositiveStep]$ .
- If the adjusted prefetch distance is greater than the prefetch distance defined for the current confidence level, the prefetch distance for the current confidence level is used.

The adaptive prefetch scheme supports fractional prefetch distances by alternating between two whole number prefetch distances. For example a prefetch distance of 1.25 causes a prefetch distance sequence of: 1, 1, 1, 2, 1, 1, 1, 2.

Bits	Description
31	<b>MctScrubEn: MCT scrub enable.</b> Read-write. Reset: 0. 1=Enables periodic flushing of prefetches and writes based on the DRAM scrub rate. This is used to ensure that prefetch and write data aging is not so long that soft errors accumulate and become uncorrectable. When enabled, each DRAM scrub event causes a single prefetch to be de-allocated (the oldest one) and all queued writes to be flushed to DRAM.
30	<b>FlushWr: flush writes command.</b> Read; write-1-only; cleared-by-hardware. Reset: 0. Setting this bit causes write bursting to be canceled and all outstanding writes to be flushed to DRAM. This bit is cleared when all writes are flushed to DRAM.
29	<b>FlushWrOnStpGnt: flush writes on stop-grant.</b> Read-write. Reset: 0. BIOS: $\sim D18F2x1B4[FlushWrOnS3StpGnt]$ . 1=Causes write bursting to be canceled and all outstanding writes to be flushed to DRAM when in the stop-grant state.
28	Reserved.
27:25	<b>PrefThreeConf: prefetch three-ahead confidence.</b> Read-write. Reset: 100b. BIOS: 110b. Confidence level required in order to prefetch three cachelines ahead (same encoding as PrefTwoConf below).

24:22	<p><b>PrefTwoConf: prefetch two-ahead confidence.</b> Read-write. Reset: 011b. BIOS: 011b. Confidence level required in order to prefetch two cachelines ahead.</p> <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>0</td> </tr> <tr> <td>110b-001b</td> <td>[PrefTwoConf*2]</td> </tr> <tr> <td>111b</td> <td>14</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Description</u>	000b	0	110b-001b	[PrefTwoConf*2]	111b	14		
<u>Bits</u>	<u>Description</u>										
000b	0										
110b-001b	[PrefTwoConf*2]										
111b	14										
21:20	<p><b>PrefOneConf: prefetch one-ahead confidence.</b> Read-write. Reset: 10b. BIOS: 10b. Confidence level required in order to prefetch one ahead (0 through 3).</p>										
19:18	<p><b>PrefConfSat: prefetch confidence saturation.</b> Read-write. Reset: 00b. BIOS: 00b. Specifies the point at which prefetch confidence level saturates and stops incrementing.</p> <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>15</td> </tr> <tr> <td>01b</td> <td>7</td> </tr> <tr> <td>10b</td> <td>3</td> </tr> <tr> <td>11b</td> <td>Reserved</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Description</u>	00b	15	01b	7	10b	3	11b	Reserved
<u>Bits</u>	<u>Description</u>										
00b	15										
01b	7										
10b	3										
11b	Reserved										
17:16	<p><b>PrefFixDist: prefetch fixed stride distance.</b> Read-write. Reset: 00b. Specifies the distance to prefetch ahead if in fixed stride mode. 00b=1 cacheline; 01b=2 cachelines; 10b=3 cachelines; 11b=4 cachelines.</p>										
15	<p><b>PrefFixStrideEn: prefetch fixed stride enable.</b> Read-write. Reset: 0. 1=The prefetch stride for all requests (CPU and IO) is fixed (non-alternating).</p>										
14	<p><b>PrefIoFixStrideEn: Prefetch IO fixed stride enable.</b> Read-write. Reset: 0. 1=The prefetch stride for IO requests is fixed (non-alternating).</p>										
13	<p><b>PrefIoDis: prefetch IO-access disable.</b> Read-write. Reset: 1. BIOS: 0. 1=Disables IO requests from triggering prefetch requests.</p>										
12	<p><b>PrefCpuDis: prefetch CPU-access disable.</b> Read-write. Reset: 1. BIOS: 0. 1=Disables CPU requests from triggering prefetch requests.</p>										
11:7	<p><b>MctPrefReqLimit: memory controller prefetch request limit.</b> Read-write. Reset: 1Eh. Specifies the maximum number of outstanding prefetch requests allowed. See <a href="#">D18F3x78</a> for restrictions on this field.</p>										
6:2	<p><b>MctWrLimit: memory controller write-burst limit.</b> Read-write. Reset: 1Fh. BIOS:0Ch. Specifies the number of writes in the memory controller queue before they are burst into the DCTs.</p> <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>00h</td> <td>32</td> </tr> <tr> <td>1Dh-01h</td> <td>[32-MctWrLimit]</td> </tr> <tr> <td>1Eh</td> <td>2</td> </tr> <tr> <td>1Fh</td> <td>Write bursting disabled</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Description</u>	00h	32	1Dh-01h	[32-MctWrLimit]	1Eh	2	1Fh	Write bursting disabled
<u>Bits</u>	<u>Description</u>										
00h	32										
1Dh-01h	[32-MctWrLimit]										
1Eh	2										
1Fh	Write bursting disabled										
1:0	<p><b>DctWrLimit: DRAM controller write limit.</b> Read-write. Reset: 00b. BIOS: 00b. Specifies the maximum number of writes allowed in the DCT queue when write bursting is enabled, prior to when the number of writes in MCQ exceeds the watermark specified by MctWrLimit.</p> <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>0</td> </tr> <tr> <td>01b</td> <td>2</td> </tr> <tr> <td>10b</td> <td>4</td> </tr> <tr> <td>11b</td> <td>Reserved</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Description</u>	00b	0	01b	2	10b	4	11b	Reserved
<u>Bits</u>	<u>Description</u>										
00b	0										
01b	2										
10b	4										
11b	Reserved										

**D18F2x1B0 Extended Memory Controller Configuration Low**

The main function of this register is to control the memory prefetcher. See [D18F2x11C \[Memory Controller Configuration High\]](#) about the adaptive prefetch scheme.

**Table 165: BIOS Recommendations** for D18F2x1B[4:0]

Condition	D18F2x1B0		D18F2x1B4	
	DdrRate	DcqBwThrotWm	DcqBwThrotWm1	DcqBwThrotWm2
667	0h	3h	4h	
800	0h	3h	5h	
1066	0h	4h	6h	
1333	0h	5h	8h	
1600	0h	6h	9h	
1866	0h	7h	Ah	
2133	0h	8h	Ch	

Bits	Description								
31:28	<b>DcqBwThrotWm: dcq bandwidth throttle watermark.</b> Read-write. Reset: 3h. BIOS: <a href="#">Table 165</a> . Specifies the number of outstanding DRAM read requests before new DRAM prefetch requests and speculative prefetch requests are throttled. 0h=Throttling is disabled. Legal values are 0h through Ch. Programming this field to a non-zero value disables <a href="#">D18F2x1B4[DcqBwThrotWm1, DcqBwThrotWm2]</a> .								
27:25	<b>PrefFiveConf: prefetch five-ahead confidence.</b> Read-write. Reset: 110b. BIOS: 111b. Confidence level required in order to prefetch five cachelines ahead. <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>0</td> </tr> <tr> <td>110b-001b</td> <td>[PrefFiveConf*2]</td> </tr> <tr> <td>111b</td> <td>14</td> </tr> </tbody> </table>	Bits	Description	000b	0	110b-001b	[PrefFiveConf*2]	111b	14
Bits	Description								
000b	0								
110b-001b	[PrefFiveConf*2]								
111b	14								
24:22	<b>PrefFourConf: prefetch four-ahead confidence.</b> Read-write. Reset: 101b. BIOS: 111b. Confidence level required in order to prefetch four cachelines ahead. <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>0</td> </tr> <tr> <td>110b-001b</td> <td>[PrefFourConf*2]</td> </tr> <tr> <td>111b</td> <td>14</td> </tr> </tbody> </table>	Bits	Description	000b	0	110b-001b	[PrefFourConf*2]	111b	14
Bits	Description								
000b	0								
110b-001b	[PrefFourConf*2]								
111b	14								
21	Reserved.								
20	<b>DbIPrefEn: double prefetch enable.</b> Read-write. Reset: 0. 1=The memory prefetcher only generates prefetch requests when it is able to generate a pair of prefetch requests to consecutive cache lines.								
19:18	Reserved.								
17:13	Reserved. Reset: 11100b.								
12	<b>EnSplitDctLimits: split DCT write limits enable.</b> Read-write. Reset: 0. BIOS: 1. 1=The number of writes specified by <a href="#">D18F2x11C[DctWrLimit, MctWrLimit]</a> is per DCT. 0=The number of writes specified by <a href="#">D18F2x11C[DctWrLimit, MctWrLimit]</a> is for the even[0,2] or odd[1,3] DCT channels. 0=The number of writes specified by <a href="#">D18F2x11C[DctWrLimit, MctWrLimit]</a> is total writes independent of DCT. Setting this bit also affects the encoding of <a href="#">D18F2x11C[DctWrLimit]</a> .								

11	<b>DisIoCohPref: disable coherent prefetched for IO.</b> Read-write. Reset: 0. 1=Probes are not generated for prefetches generated for reads from IO devices.										
10:8	<b>CohPrefPrbLmt: coherent prefetch probe limit.</b> Read-write. Reset: 000b. BIOS: 000b. Specifies the maximum number of probes that can be outstanding for memory prefetch requests. <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>Probing disabled for memory prefetch requests</td> </tr> <tr> <td>111b-001b</td> <td>Reserved.</td> </tr> </tbody> </table>	Bits	Description	000b	Probing disabled for memory prefetch requests	111b-001b	Reserved.				
Bits	Description										
000b	Probing disabled for memory prefetch requests										
111b-001b	Reserved.										
7:6	Reserved.										
5:4	<b>AdapPrefNegativeStep: adaptive prefetch negative step.</b> Read-write. Reset: 00b. BIOS: 00b. Specifies the step size that the adaptive prefetch scheme uses when decreasing the prefetch distance. <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>2/16</td> </tr> <tr> <td>01b</td> <td>4/16</td> </tr> <tr> <td>10b</td> <td>8/16</td> </tr> <tr> <td>11b</td> <td>16/16</td> </tr> </tbody> </table>	Bits	Description	00b	2/16	01b	4/16	10b	8/16	11b	16/16
Bits	Description										
00b	2/16										
01b	4/16										
10b	8/16										
11b	16/16										
3:2	<b>AdapPrefPositiveStep: adaptive prefetch positive step.</b> Read-write. Reset: 00b. BIOS: 00b. Specifies the step size that the adaptive prefetch scheme uses when increasing the prefetch distance. <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>1/16</td> </tr> <tr> <td>01b</td> <td>2/16</td> </tr> <tr> <td>10b</td> <td>4/16</td> </tr> <tr> <td>11b</td> <td>8/16</td> </tr> </tbody> </table>	Bits	Description	00b	1/16	01b	2/16	10b	4/16	11b	8/16
Bits	Description										
00b	1/16										
01b	2/16										
10b	4/16										
11b	8/16										
1:0	<b>AdapPrefMissRatio: adaptive prefetch miss ratio.</b> Read-write. Reset: 00b. BIOS: 01b. Specifies the ratio of prefetch requests that do not have data buffer available to the total number of prefetch requests at which the adaptive prefetch scheme begins decreasing the prefetch distance. <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>1/16</td> </tr> <tr> <td>01b</td> <td>2/16</td> </tr> <tr> <td>10b</td> <td>4/16</td> </tr> <tr> <td>11b</td> <td>8/16</td> </tr> </tbody> </table>	Bits	Description	00b	1/16	01b	2/16	10b	4/16	11b	8/16
Bits	Description										
00b	1/16										
01b	2/16										
10b	4/16										
11b	8/16										

### D18F2x1B4 Extended Memory Controller Configuration High Register

Bits	Description
31	<b>FlushOnMmioWrEn: flush on mmio write enable.</b> Read-write. Reset: 0. 1=Any CPU-sourced MMIO write that matches <a href="#">D18F1x[2CC:2A0,1CC:180,BC:80]</a> causes the memory controller data buffers to be flushed to memory.
30:28	<b>S3SmafId: S3 SMAF id.</b> Read-write. Reset: 100b. SMAF encoding of <a href="#">D18F3x[84:80]</a> corresponding to the ACPI S3 state when FlushWrOnS3StpGnt=1. Reserved when FlushWrOnS3StpGnt=0.
27	<b>FlushWrOnS3StpGnt: flush write on S3 stop grant.</b> Read-write. Reset: 0. BIOS: 1. 1=Write bursting is canceled and all outstanding writes are flushed to DRAM when in the stop-grant state and the SMAF code is equal to S3SmafId, indicating entry into the ACPI S3 state. See <a href="#">D18F2xA8_dct[0][FastSelfRefEntryDis]</a> , <a href="#">D18F2x11C[FlushWrOnStpGnt]</a> .
26	<b>EnSplitMctDatBuffers: enable split MCT data buffers.</b> Read-write. Reset: 0. BIOS: 1. 1=Enable resource allocation into the split buffer resources BIOS must program this bit before any DRAM memory accesses are issued from the processor.

25	<b>SmuCfgLock: SMU configuration lock.</b> Read-write; updated-by-hardware. Reset: 0. This field should never be cleared by software. The following registers are read-only if LockSmuCfg=1; otherwise the access type is specified by the register: <ul style="list-style-type: none"> <li>• D18F4x15C [Core Performance Boost Control]</li> <li>• D18F5x12C [Clock Power/Timing Control 4]</li> <li>• D18F5x170 [Northbridge P-state Control]</li> <li>• D18F5x188 [Clock Power/Timing Control 5]</li> </ul>
24:23	Reserved.
22	<b>SpecPrefDisWm1: speculative prefetch disable watermark 1.</b> Read-write. Reset: 0. 0=Disable speculative prefetches at the DcqBwThrotWm2 limit. 1=Disable speculative prefetches at the DcqBwThrotWm1 limit. See also D18F2x1B0[SpecPrefDis].
21	<b>RegionAlloWm2: region prefetch allocate watermark 2.</b> Read-write. Reset: 0. See DemandAlloWm2.
20	<b>RegionPropWm2: region prefetch propagate watermark 2.</b> Read-write. Reset: 0. See DemandPropWm2.
19	<b>StrideAlloWm2: stride prefetch allocate watermark 2.</b> Read-write. Reset: 1. See DemandAlloWm2.
18	<b>StridePropWm2: stride prefetch propagate watermark 2.</b> Read-write. Reset: 1. See DemandPropWm2.
17	<b>DemandAlloWm2: demand request allocate watermark 2.</b> Read-write. Reset: 1. Specifies the behavior from the DcqBwThrotWm1 limit to the DcqBwThrotWm2 limit. 0=Requests do not allocate a new entry. 1=Requests allocate a new entry; defined only if (DemandAlloWm1 & DemandPropWm2).
16	<b>DemandPropWm2: demand request propagate watermark 2.</b> Read-write. Reset: 1. Specifies the behavior from the DcqBwThrotWm1 limit to the DcqBwThrotWm2 limit. 0=Requests do not update existing entries. 1=Requests update existing entries; defined only if (DemandPropWm1=1).
15	<b>RegionAlloWm1: region prefetch allocate watermark 1.</b> Read-write. Reset: 0. See DemandAlloWm1.
14	<b>RegionPropWm1: region prefetch propagate watermark 1.</b> Read-write. Reset: 1. See DemandPropWm1.
13	<b>StrideAlloWm1: stride prefetch allocate watermark 1.</b> Read-write. Reset: 1. See DemandAlloWm1.
12	<b>StridePropWm1: stride prefetch propagate watermark 1.</b> Read-write. Reset: 1. See DemandPropWm1.
11	<b>DemandAlloWm1: demand request allocate watermark 1.</b> Read-write. Reset: 1. Specifies the behavior prior to the DcqBwThrotWm1 limit. 0=Requests do not allocate a new entry. 1=Requests allocate a new entry; defined only if (DemandPropWm1=1).
10	<b>DemandPropWm1: demand request propagate watermark 1.</b> Read-write. Reset: 1. Specifies the behavior prior to the DcqBwThrotWm1 limit. 0=Requests do not update existing entries. 1=Requests update existing entries.

9:5	<b>DcqBwThrotWm2: DCQ bandwidth throttle watermark 2.</b> Read-write. Reset: 06h. BIOS: <a href="#">Table 165</a> . Specifies a prefetch throttling watermark based on the number of outstanding DRAM read requests. This field is reserved when <code>D18F2x1B0[DcqBwThrotWm] != 0</code> . When throttling is enabled, if the number of outstanding DRAM read requests exceeds DcqBwThrotWm2 both request allocate and propagate are blocked and new prefetches are disabled. When throttling is enabled, DcqBwThrotWm2 should be programmed to a value greater than DcqBwThrotWm1. 0h=Throttling is disabled. Legal values are 0h through 18h.
4:0	<b>DcqBwThrotWm1: DCQ bandwidth throttle watermark 1.</b> Read-write. Reset: 03h. BIOS: <a href="#">Table 165</a> . Specifies a prefetch throttling watermark based on the number of outstanding DRAM read requests. This field is reserved when <code>D18F2x1B0[DcqBwThrotWm] != 0</code> . 0h=Throttling is disabled. Legal values are 0h through 18h.

**D18F2x1BC\_dct[0] DRAM CKE to CS Map**

Reset: 0000\_AA55h. See [2.9.3 \[DCT Configuration Registers\]](#).

Table 166: Field Mapping for [D18F2x1BC\\_dct\[0\]](#)

Register	Bits			
	31:24	23:16	15:8	7:0
<a href="#">D18F2x1BC_dct[0]</a>	CKE3	CKE2	CKE1	CKE0

Table 167: BIOS Recommendations for [D18F2x1BC\\_dct\[0\]](#)

Condition:		<a href="#">D18F2x1BC_dct[0]</a>
Package	NumDimmSlots	
FT3	1, 2	08040201h

Bits	Description												
31:24	<b>CSMapCKE: CS map CKE.</b> See: <a href="#">D18F2x1BC_dct[0][7:0]</a> .												
23:16	<b>CSMapCKE: CS map CKE.</b> See: <a href="#">D18F2x1BC_dct[0][7:0]</a> .												
15:8	<b>CSMapCKE: CS map CKE.</b> See: <a href="#">D18F2x1BC_dct[0][7:0]</a> .												
7:0	<p><b>CSMapCKE: CS map CKE.</b> Read-write. BIOS: See <a href="#">Table 167 [BIOS Recommendations for D18F2x1BC_dct[0]]</a>. Maps the CS to CKE relationship. 1=This CKE is associated with the listed chip select. 0=This CKE is not associated with the listed chip select. See also <a href="#">D18F2x9C_x0000_000C_dct[0][CKETri]</a>.</p> <p>Only 1 CKE may be assigned to a CS across <a href="#">D18F2x1BC_dct[0]</a>. Only even CKEs may be assigned to even CSes. Only odd CKEs may be assigned to odd CSes.</p> <table border="1"> <thead> <tr> <th>Bit</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>[0]</td> <td>CS0</td> </tr> <tr> <td>[1]</td> <td>CS1</td> </tr> <tr> <td>[2]</td> <td>CS2</td> </tr> <tr> <td>[3]</td> <td>CS3</td> </tr> <tr> <td>[7:4]</td> <td>Reserved</td> </tr> </tbody> </table>	Bit	Description	[0]	CS0	[1]	CS1	[2]	CS2	[3]	CS3	[7:4]	Reserved
Bit	Description												
[0]	CS0												
[1]	CS1												
[2]	CS2												
[3]	CS3												
[7:4]	Reserved												



**D18F2x200\_dct[0]\_mp[1:0] DDR3 DRAM Timing 0**

Reset: 0F05\_0505h. See 2.9.3 [DCT Configuration Registers].

Bits	Description								
31:30	Reserved.								
29:24	<p><b>Tras: row active strobe.</b> Read-write. BIOS: See 2.9.7.5 [SPD ROM-Based Configuration]. Specifies the minimum time in memory clock cycles from an activate command to a precharge command, both to the same chip select bank.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>07h-00h</td> <td>Reserved</td> </tr> <tr> <td>2Ah-08h</td> <td>&lt;Tras&gt; clocks</td> </tr> <tr> <td>3Fh-2Bh</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Description	07h-00h	Reserved	2Ah-08h	<Tras> clocks	3Fh-2Bh	Reserved
Bits	Description								
07h-00h	Reserved								
2Ah-08h	<Tras> clocks								
3Fh-2Bh	Reserved								
23:21	Reserved.								
20:16	<p><b>Trp: row precharge time.</b> Read-write. BIOS: See 2.9.7.5 [SPD ROM-Based Configuration]. 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.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>04h-00h</td> <td>Reserved</td> </tr> <tr> <td>13h-05h</td> <td>&lt;Trp&gt; clocks</td> </tr> <tr> <td>1Fh-14h</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Description	04h-00h	Reserved	13h-05h	<Trp> clocks	1Fh-14h	Reserved
Bits	Description								
04h-00h	Reserved								
13h-05h	<Trp> clocks								
1Fh-14h	Reserved								
15:13	Reserved.								
12:8	<p><b>Trcd: RAS to CAS delay.</b> Read-write. BIOS: See 2.9.7.5 [SPD ROM-Based Configuration]. Specifies the time in memory clock cycles from an activate command to a read/write command, both to the same bank.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>04h-00h</td> <td>Reserved</td> </tr> <tr> <td>13h-05h</td> <td>&lt;Trcd&gt; clocks</td> </tr> <tr> <td>1Fh-14h</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Description	04h-00h	Reserved	13h-05h	<Trcd> clocks	1Fh-14h	Reserved
Bits	Description								
04h-00h	Reserved								
13h-05h	<Trcd> clocks								
1Fh-14h	Reserved								
7:5	Reserved.								
4:0	<p><b>Tcl: CAS latency.</b> Read-write. BIOS: See 2.9.7.5 [SPD ROM-Based Configuration]. Specifies the time in memory clock cycles from the CAS assertion for a read cycle until data return (from the perspective of the DRAM devices).</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>04h-00h</td> <td>Reserved</td> </tr> <tr> <td>13h-05h</td> <td>&lt;Tcl&gt; clocks</td> </tr> <tr> <td>1Fh-14h</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Description	04h-00h	Reserved	13h-05h	<Tcl> clocks	1Fh-14h	Reserved
Bits	Description								
04h-00h	Reserved								
13h-05h	<Tcl> clocks								
1Fh-14h	Reserved								

**D18F2x204\_dct[0]\_mp[1:0] DDR3 DRAM Timing 1**

Reset: 0400\_040Bh. See 2.9.3 [DCT Configuration Registers].

Bits	Description
31:28	Reserved.

27:24	<b>Trtp: read CAS to precharge time.</b> Read-write. BIOS: See 2.9.7.5. 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. <table border="0"> <thead> <tr> <th><u>Bits</u></th> <th><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>3h-0h</td> <td>Reserved</td> </tr> <tr> <td>Bh-4h</td> <td>&lt;Trtp&gt; clocks</td> </tr> <tr> <td>Fh-Ch</td> <td>Reserved</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Description</u>	3h-0h	Reserved	Bh-4h	<Trtp> clocks	Fh-Ch	Reserved		
<u>Bits</u>	<u>Description</u>										
3h-0h	Reserved										
Bh-4h	<Trtp> clocks										
Fh-Ch	Reserved										
23:22	Reserved.										
21:16	<b>FourActWindow: four bank activate window.</b> Read-write. BIOS: See 2.9.7.5. Specifies the rolling tFAW window in memory clock cycles during which no more than 4 banks in an 8-bank device are activated. <table border="0"> <thead> <tr> <th><u>Bits</u></th> <th><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>00h</td> <td>No tFAW window restriction</td> </tr> <tr> <td>05h-01h</td> <td>Reserved</td> </tr> <tr> <td>2Ch-06h</td> <td>[FourActWindow] clocks</td> </tr> <tr> <td>3Fh-2Dh</td> <td>Reserved</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Description</u>	00h	No tFAW window restriction	05h-01h	Reserved	2Ch-06h	[FourActWindow] clocks	3Fh-2Dh	Reserved
<u>Bits</u>	<u>Description</u>										
00h	No tFAW window restriction										
05h-01h	Reserved										
2Ch-06h	[FourActWindow] clocks										
3Fh-2Dh	Reserved										
15:12	Reserved.										
11:8	<b>Trrd: row to row delay (or RAS to RAS delay).</b> Read-write. BIOS: See 2.9.7.5. Specifies the minimum time in memory clock cycles between activate commands to different chip select banks. <table border="0"> <thead> <tr> <th><u>Bits</u></th> <th><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Reserved</td> </tr> <tr> <td>9h-1h</td> <td>&lt;Trrd&gt; clocks</td> </tr> <tr> <td>Fh-Ah</td> <td>Reserved</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Description</u>	0h	Reserved	9h-1h	<Trrd> clocks	Fh-Ah	Reserved		
<u>Bits</u>	<u>Description</u>										
0h	Reserved										
9h-1h	<Trrd> clocks										
Fh-Ah	Reserved										
7:6	Reserved.										
5:0	<b>Trc: row cycle time.</b> Read-write. BIOS: See 2.9.7.5. Specifies the minimum time in memory clock cycles from and activate command to another activate command or an auto refresh command, all to the same chip select bank. <table border="0"> <thead> <tr> <th><u>Bits</u></th> <th><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>09h-00h</td> <td>Reserved</td> </tr> <tr> <td>3Ah-0Ah</td> <td>&lt;Trc&gt; clocks</td> </tr> <tr> <td>3Fh-3Bh</td> <td>Reserved</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Description</u>	09h-00h	Reserved	3Ah-0Ah	<Trc> clocks	3Fh-3Bh	Reserved		
<u>Bits</u>	<u>Description</u>										
09h-00h	Reserved										
3Ah-0Ah	<Trc> clocks										
3Fh-3Bh	Reserved										

**D18F2x208\_dct[0] DDR3 DRAM Timing 2**

See 2.9.3 [DCT Configuration Registers].

Bits	Description
31:27	Reserved.
26:24	<b>Trfc3: auto refresh row cycle time for CS 6 and 7.</b> See: Trfc0.
23:19	Reserved.
18:16	<b>Trfc2: auto refresh row cycle time for CS 4 and 5.</b> See: Trfc0.
15:11	Reserved.
10:8	<b>Trfc1: auto refresh row cycle time for CS 2 and 3.</b> See: Trfc0.

7:3	Reserved.																
2:0	<p><b>Trfc0: auto refresh row cycle time for CS 0 and 1.</b> Read-write. Reset: 100b. BIOS: <a href="#">2.9.7.5</a>. Specifies the minimum time from a refresh command to the next valid command, except NOP or DES. The recommended programming of this register varies based on DRAM density and speed.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>Reserved</td> </tr> <tr> <td>001b</td> <td>90 ns (all speeds, 512 Mbit)</td> </tr> <tr> <td>010b</td> <td>110 ns (all speeds, 1 Gbit)</td> </tr> <tr> <td>011b</td> <td>160 ns (all speeds, 2 Gbit)</td> </tr> <tr> <td>100b</td> <td>300 ns (all speeds, 4 Gbit)</td> </tr> <tr> <td>101b</td> <td>350 ns (all speeds, 8 Gbit)</td> </tr> <tr> <td>111b-110b</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Description	000b	Reserved	001b	90 ns (all speeds, 512 Mbit)	010b	110 ns (all speeds, 1 Gbit)	011b	160 ns (all speeds, 2 Gbit)	100b	300 ns (all speeds, 4 Gbit)	101b	350 ns (all speeds, 8 Gbit)	111b-110b	Reserved
Bits	Description																
000b	Reserved																
001b	90 ns (all speeds, 512 Mbit)																
010b	110 ns (all speeds, 1 Gbit)																
011b	160 ns (all speeds, 2 Gbit)																
100b	300 ns (all speeds, 4 Gbit)																
101b	350 ns (all speeds, 8 Gbit)																
111b-110b	Reserved																

### D18F2x20C\_dct[0]\_mp[1:0] DDR3 DRAM Timing 3

See [2.9.3 \[DCT Configuration Registers\]](#).

Bits	Description										
31:18	Reserved.										
17:16	<p><b>WrDqDqsEarly: DQ and DQS write early.</b> Read-Write. Reset: 0. Specifies the DQ and DQS launch timing for write commands relative to the Tcwl MEMCLK.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>0 MEMCLK early</td> </tr> <tr> <td>01b</td> <td>0.5 MEMCLK early</td> </tr> <tr> <td>10b</td> <td>Reserved</td> </tr> <tr> <td>11b</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Description	00b	0 MEMCLK early	01b	0.5 MEMCLK early	10b	Reserved	11b	Reserved
Bits	Description										
00b	0 MEMCLK early										
01b	0.5 MEMCLK early										
10b	Reserved										
11b	Reserved										
15:12	Reserved.										
11:8	<p><b>Twtr: internal DRAM write to read command delay.</b> Read-write. Reset: 4h. BIOS: See <a href="#">2.9.7.5</a>. 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 last non-masked data strobe of the write to the rising clock edge of the next read command.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>3h-0h</td> <td>Reserved</td> </tr> <tr> <td>Bh-4h</td> <td>&lt;Twtr&gt; clocks</td> </tr> <tr> <td>Fh-Ch</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Description	3h-0h	Reserved	Bh-4h	<Twtr> clocks	Fh-Ch	Reserved		
Bits	Description										
3h-0h	Reserved										
Bh-4h	<Twtr> clocks										
Fh-Ch	Reserved										
7:5	Reserved.										
4:0	<p><b>Tcwl: CAS write latency.</b> Read-write. Reset: 5h. BIOS: See <a href="#">2.9.7.5</a>. Specifies the number of memory clock cycles from internal write command to first write data in at the DRAM.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>04h-00h</td> <td>Reserved</td> </tr> <tr> <td>0Ah-05h</td> <td>&lt;Tcwl&gt; clocks</td> </tr> <tr> <td>1Fh-0Bh</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Description	04h-00h	Reserved	0Ah-05h	<Tcwl> clocks	1Fh-0Bh	Reserved		
Bits	Description										
04h-00h	Reserved										
0Ah-05h	<Tcwl> clocks										
1Fh-0Bh	Reserved										

### D18F2x210\_dct[0]\_nbp[3:0] DRAM NB P-state

See [2.9.3 \[DCT Configuration Registers\]](#). For D18F2x210\_dct[0]\_nbp[x], x=D18F1x10C[NbPsSel]; see D18F1x10C[NbPsSel].

**Table 168: BIOS Recommendations for RdPtrInit**

Condition		D18F2x210_dct[0]_nbp[3:0]
NBCOF >= DdrRate	DdrRate	RdPtrInit
0	-	0010b
1	667, 800, 1066, 1333, 1600	0110b
1	1866, 2133	0101b

Bits	Description																		
31:22	<p><b>MaxRdLatency: maximum read latency.</b> Read-write. Reset: 000h. BIOS: See 2.9.7.9.5 [Calculating MaxRdLatency]. Specifies the maximum round-trip latency in the system from the processor to the DRAM devices and back. The DRAM controller uses this to help determine when the first two beats of incoming DRAM read data can be safely transferred to the NCLK domain. The time includes the asynchronous and synchronous latencies.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>000h</td> <td>0 NCLKs</td> </tr> <tr> <td>3FEh-001h</td> <td>&lt;MaxRdLatency&gt; NCLKs</td> </tr> <tr> <td>3FFh</td> <td>1023 NCLKs</td> </tr> </tbody> </table>	Bits	Description	000h	0 NCLKs	3FEh-001h	<MaxRdLatency> NCLKs	3FFh	1023 NCLKs										
Bits	Description																		
000h	0 NCLKs																		
3FEh-001h	<MaxRdLatency> NCLKs																		
3FFh	1023 NCLKs																		
21:19	Reserved.																		
18:16	<p><b>DataTxFifoWrDly: data transmit FIFO write delay.</b> Read-write. Reset: 0. BIOS: 0h. Specifies the DCT to phy write data FIFO delay.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>0 MEMCLK</td> </tr> <tr> <td>001b</td> <td>0.5 MEMCLK</td> </tr> <tr> <td>010b</td> <td>1.0 MEMCLK</td> </tr> <tr> <td>011b</td> <td>1.5 MEMCLKs</td> </tr> <tr> <td>100b</td> <td>2.0 MEMCLKs</td> </tr> <tr> <td>101b</td> <td>2.5 MEMCLKs</td> </tr> <tr> <td>110b</td> <td>3.0 MEMCLKs</td> </tr> <tr> <td>111b</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Description	000b	0 MEMCLK	001b	0.5 MEMCLK	010b	1.0 MEMCLK	011b	1.5 MEMCLKs	100b	2.0 MEMCLKs	101b	2.5 MEMCLKs	110b	3.0 MEMCLKs	111b	Reserved
Bits	Description																		
000b	0 MEMCLK																		
001b	0.5 MEMCLK																		
010b	1.0 MEMCLK																		
011b	1.5 MEMCLKs																		
100b	2.0 MEMCLKs																		
101b	2.5 MEMCLKs																		
110b	3.0 MEMCLKs																		
111b	Reserved																		
15:4	Reserved.																		
3:0	<p><b>RdPtrInit: read pointer initial value.</b> Read-write. Reset: 6h. BIOS: Table 168. There is a synchronization FIFO between the NB clock domain and memory clock domain. Each increment of this field positions the read pointer one half clock cycle closer to the write pointer thereby reducing the latency through the FIFO. BIOS must program this field for the current or target NB P-state prior to a frequency change or NB P-state change.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0001b-0000b</td> <td>Reserved</td> </tr> <tr> <td>0010b</td> <td>3 MEMCLK</td> </tr> <tr> <td>0011b</td> <td>2.5 MEMCLK</td> </tr> <tr> <td>0100b</td> <td>2 MEMCLKs</td> </tr> <tr> <td>0101b</td> <td>1.5 MEMCLKs</td> </tr> <tr> <td>0110b</td> <td>1 MEMCLKs</td> </tr> <tr> <td>1111b-0111b</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Description	0001b-0000b	Reserved	0010b	3 MEMCLK	0011b	2.5 MEMCLK	0100b	2 MEMCLKs	0101b	1.5 MEMCLKs	0110b	1 MEMCLKs	1111b-0111b	Reserved		
Bits	Description																		
0001b-0000b	Reserved																		
0010b	3 MEMCLK																		
0011b	2.5 MEMCLK																		
0100b	2 MEMCLKs																		
0101b	1.5 MEMCLKs																		
0110b	1 MEMCLKs																		
1111b-0111b	Reserved																		

**D18F2x214\_dct[0]\_mp[1:0] DDR3 DRAM Timing 4**

Reset: 0001\_0202h.

Bits	Description												
31:20	Reserved.												
19:16	<p><b>TwrrwrSdSc: write to write timing same DIMM same chip select.</b> Read-write. BIOS: See 2.9.7.6.2 [TwrrwrSdSc, TwrrwrSdDc, TwrrwrDd (Write to Write Timing)]. 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.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Reserved</td> </tr> <tr> <td>1h</td> <td>1 clock</td> </tr> <tr> <td>Ah-2h</td> <td>&lt;TwrrwrSdSc&gt; clocks</td> </tr> <tr> <td>Bh</td> <td>11 clocks</td> </tr> <tr> <td>Fh-Ch</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Description	0h	Reserved	1h	1 clock	Ah-2h	<TwrrwrSdSc> clocks	Bh	11 clocks	Fh-Ch	Reserved
Bits	Description												
0h	Reserved												
1h	1 clock												
Ah-2h	<TwrrwrSdSc> clocks												
Bh	11 clocks												
Fh-Ch	Reserved												
15:12	Reserved.												
11:8	<b>TwrrwrSdDc: write to write timing same DIMM different chip select.</b> See: TwrrwrDd.												
7:4	Reserved.												
3:0	<p><b>TwrrwrDd: write to write timing different DIMM.</b> Read-write. BIOS: See 2.9.7.6.2 [TwrrwrSdSc, TwrrwrSdDc, TwrrwrDd (Write to Write Timing)]. 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.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>1h-0h</td> <td>Reserved</td> </tr> <tr> <td>Bh-2h</td> <td>&lt;TwrrwrDd&gt; clocks</td> </tr> <tr> <td>Fh-Ch</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Description	1h-0h	Reserved	Bh-2h	<TwrrwrDd> clocks	Fh-Ch	Reserved				
Bits	Description												
1h-0h	Reserved												
Bh-2h	<TwrrwrDd> clocks												
Fh-Ch	Reserved												

**D18F2x218\_dct[0]\_mp[1:0] DDR3 DRAM Timing 5**

Reset: 0103\_0203h. See 2.9.3 [DCT Configuration Registers].

Bits	Description										
31:30	<p><b>TrdrdBan: read to read timing ban.</b> Read-write. BIOS: 00b. Bans the traffic for the specified cases where the 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.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Ban disabled, traffic allowed as specified by TrdrdSdSc, TrdrdSdDc, TrdrdDd.</td> </tr> <tr> <td>01b</td> <td>Ban Trdrd traffic at 2 MEMCLKs.</td> </tr> <tr> <td>10b</td> <td>Ban Trdrd traffic at 2 and 3 MEMCLKs.</td> </tr> <tr> <td>11b</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Description	00b	Ban disabled, traffic allowed as specified by TrdrdSdSc, TrdrdSdDc, TrdrdDd.	01b	Ban Trdrd traffic at 2 MEMCLKs.	10b	Ban Trdrd traffic at 2 and 3 MEMCLKs.	11b	Reserved
Bits	Description										
00b	Ban disabled, traffic allowed as specified by TrdrdSdSc, TrdrdSdDc, TrdrdDd.										
01b	Ban Trdrd traffic at 2 MEMCLKs.										
10b	Ban Trdrd traffic at 2 and 3 MEMCLKs.										
11b	Reserved										
29:28	Reserved.										

27:24	<p><b>TrdrdSdSc: read to read timing same DIMM same chip select.</b> Read-write. BIOS: See <a href="#">2.9.7.6.1 [TrdrdSdSc, TrdrdSdDc, and TrdrdDd (Read to Read Timing)]</a>. 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.</p> <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Reserved</td> </tr> <tr> <td>Bh-1h</td> <td>&lt;TrdrdSdSc&gt; clocks</td> </tr> <tr> <td>Fh-Ch</td> <td>Reserved</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Description</u>	0h	Reserved	Bh-1h	<TrdrdSdSc> clocks	Fh-Ch	Reserved
<u>Bits</u>	<u>Description</u>								
0h	Reserved								
Bh-1h	<TrdrdSdSc> clocks								
Fh-Ch	Reserved								
23:20	Reserved.								
19:16	<b>TrdrdSdDc: read to read timing same DIMM different chip select.</b> See: TrdrdDd.								
15:12	Reserved.								
11:8	<p><b>Twrrd: write to read DIMM termination turnaround.</b> Read-write. BIOS: See <a href="#">2.9.7.6.3 [Twrrd (Write to Read DIMM Termination Turn-around)]</a>. 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 read-burst operation, both to different chip selects.</p> <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Reserved</td> </tr> <tr> <td>Bh-1h</td> <td>&lt;Twrrd&gt; clocks</td> </tr> <tr> <td>Fh-Ch</td> <td>Reserved</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Description</u>	0h	Reserved	Bh-1h	<Twrrd> clocks	Fh-Ch	Reserved
<u>Bits</u>	<u>Description</u>								
0h	Reserved								
Bh-1h	<Twrrd> clocks								
Fh-Ch	Reserved								
7:4	Reserved.								
3:0	<p><b>TrdrdDd: read to read timing different DIMM.</b> Read-write. BIOS: See <a href="#">2.9.7.6.1 [TrdrdSdSc, TrdrdSdDc, and TrdrdDd (Read to Read Timing)]</a>. 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.</p> <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>1h-0h</td> <td>Reserved</td> </tr> <tr> <td>Bh-2h</td> <td>&lt;TrdrdDd&gt; clocks</td> </tr> <tr> <td>Fh-Ch</td> <td>Reserved</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Description</u>	1h-0h	Reserved	Bh-2h	<TrdrdDd> clocks	Fh-Ch	Reserved
<u>Bits</u>	<u>Description</u>								
1h-0h	Reserved								
Bh-2h	<TrdrdDd> clocks								
Fh-Ch	Reserved								

### D18F2x21C\_dct[0]\_mp[1:0] DDR3 DRAM Timing 6

Reset: 0004\_0300h. See [2.9.3 \[DCT Configuration Registers\]](#).

Bits	Description								
31:21	Reserved.								
20:16	<p><b>TrwtWB: read to write turnaround for opportunistic write bursting.</b> Read-write. BIOS: TrwtTO + 1. Specifies the minimum number of clock 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 write-burst operation.</p> <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>02h-00h</td> <td>Reserved</td> </tr> <tr> <td>1Ch-03h</td> <td>&lt;TrwtWB&gt; clocks</td> </tr> <tr> <td>1Fh-1Dh</td> <td>Reserved</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Description</u>	02h-00h	Reserved	1Ch-03h	<TrwtWB> clocks	1Fh-1Dh	Reserved
<u>Bits</u>	<u>Description</u>								
02h-00h	Reserved								
1Ch-03h	<TrwtWB> clocks								
1Fh-1Dh	Reserved								
15:13	Reserved.								

12:8	<p><b>TrwtTO: read to write turnaround.</b> Read-write. BIOS: See 2.9.7.6.4 [TrwtTO (Read-to-Write Turnaround for Data, DQS Contention)]. Specifies the minimum number of clock 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 write-burst operation.</p> <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>01h-00h</td> <td>Reserved</td> </tr> <tr> <td>1Bh-02h</td> <td>&lt;TrwtTO&gt; clocks</td> </tr> <tr> <td>1Fh-1Ch</td> <td>Reserved</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Description</u>	01h-00h	Reserved	1Bh-02h	<TrwtTO> clocks	1Fh-1Ch	Reserved
<u>Bits</u>	<u>Description</u>								
01h-00h	Reserved								
1Bh-02h	<TrwtTO> clocks								
1Fh-1Ch	Reserved								
7:0	Reserved.								

**D18F2x220\_dct[0] DDR3 DRAM Timing 7**

Reset: 0000\_0C04h. See 2.9.3 [DCT Configuration Registers].

Bits	Description								
31:13	Reserved.								
12:8	<p><b>Tmod: mode register command delay.</b> Read-write. BIOS: See 2.9.7.5. Specifies the minimum time in memory clock cycles from an MRS command to another non-MRS command (excluding NOP and DES), all to the same chip select.</p> <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>1h-0h</td> <td>Reserved</td> </tr> <tr> <td>14h-2h</td> <td>&lt;Tmod&gt; clocks</td> </tr> <tr> <td>1Fh-15h</td> <td>Reserved</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Description</u>	1h-0h	Reserved	14h-2h	<Tmod> clocks	1Fh-15h	Reserved
<u>Bits</u>	<u>Description</u>								
1h-0h	Reserved								
14h-2h	<Tmod> clocks								
1Fh-15h	Reserved								
7:4	Reserved.								
3:0	<p><b>Tmrd: mode register command cycle time.</b> Read-write. BIOS: See 2.9.7.5. Specifies the minimum time in memory clock cycles from an MRS command to another MRS command, all to the same chip select.</p> <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>1h-0h</td> <td>Reserved</td> </tr> <tr> <td>8h-2h</td> <td>&lt;Tmrd&gt; clocks</td> </tr> <tr> <td>Fh-9h</td> <td>Reserved</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Description</u>	1h-0h	Reserved	8h-2h	<Tmrd> clocks	Fh-9h	Reserved
<u>Bits</u>	<u>Description</u>								
1h-0h	Reserved								
8h-2h	<Tmrd> clocks								
Fh-9h	Reserved								

**D18F2x224\_dct[0] DDR3 DRAM Timing 8**

Reset: 0000\_0408h. See 2.9.3 [DCT Configuration Registers].

Bits	Description																				
31:11	Reserved.																				
10:8	<p><b>Tzqcs: Zq short cal command delay.</b> Read-write. BIOS: See 2.9.7.5. Specifies the minimum time in memory clock cycles from a ZQCS command to any other command (excluding NOP and DES) on the channel.</p> <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Description</u></th> <th><u>Bits</u></th> <th><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>Reserved</td> <td>100b</td> <td>64 clocks</td> </tr> <tr> <td>001b</td> <td>16 clocks</td> <td>101b</td> <td>80 clocks</td> </tr> <tr> <td>010b</td> <td>32 clocks</td> <td>110b</td> <td>96 clocks</td> </tr> <tr> <td>011b</td> <td>48 clocks</td> <td>111b</td> <td>Reserved</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Description</u>	<u>Bits</u>	<u>Description</u>	000b	Reserved	100b	64 clocks	001b	16 clocks	101b	80 clocks	010b	32 clocks	110b	96 clocks	011b	48 clocks	111b	Reserved
<u>Bits</u>	<u>Description</u>	<u>Bits</u>	<u>Description</u>																		
000b	Reserved	100b	64 clocks																		
001b	16 clocks	101b	80 clocks																		
010b	32 clocks	110b	96 clocks																		
011b	48 clocks	111b	Reserved																		

7:4	Reserved.			
3:0	<b>Tzqoper: Zq long cal command delay.</b> Read-write. BIOS: See 2.9.7.5. Specifies the minimum time in memory clock cycles from a ZQCL command to any other command (excluding NOP and DES) on the channel.			
	<u>Bits</u>	<u>Description</u>	<u>Bits</u>	<u>Description</u>
	0000b	Reserved	1000b	256 clocks
	0001b	32 clocks	1001b	288 clocks
	0010b	64 clocks	1010b	320 clocks
	0011b	96 clocks	1011b	352 clocks
	0100b	128 clocks	1100b	384 clocks
	0101b	160 clocks	1111b-1101b	Reserved
	0110b	192 clocks		
	0111b	224 clocks		

**D18F2x228\_dct[0] DDR3 DRAM Timing 9**

See 2.9.3 [DCT Configuration Registers].

Bits	Description
31:24	<b>Tstag3: auto refresh stagger time for logical DIMM 3.</b> See: Tstag0.
23:16	<b>Tstag2: auto refresh stagger time for logical DIMM 2.</b> See: Tstag0.
15:8	<b>Tstag1: auto refresh stagger time for logical DIMM 1.</b> See: Tstag0.
7:0	<b>Tstag0: auto refresh stagger time for logical DIMM 0.</b> Read-write. Reset: 00h. BIOS: $\text{MAX}(\text{D18F2x204\_dct}[0]\_mp[1:0][\text{Trrd}], \text{CEIL}(\text{D18F2x204\_dct}[0]\_mp[1:0][\text{FourActWindow}]/4))$ . Specifies the number of clocks between auto refresh commands to different ranks of a DIMM when $\text{D18F2x90\_dct}[0][\text{StagRefEn}] = 1$ .
	<u>Bits</u> <u>Description</u>
	00h                          0 clocks
	FEh-01h                  <Tstag0> clocks
	FFh                          255 clocks

**D18F2x22C\_dct[0]\_mp[1:0] DDR3 DRAM Timing 10**

Reset: 0000\_000Ch. See 2.9.3 [DCT Configuration Registers].

Bits	Description
------	-------------



31:5	Reserved.																												
4:0	<p><b>Twr: write recovery.</b> Read-write. BIOS: See <a href="#">2.9.7.5</a>. Specifies the minimum time from the last data write until the chip select bank precharge.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>4h-0h</td> <td>Reserved</td> </tr> <tr> <td>8h-5h</td> <td>8 to 5 clocks</td> </tr> <tr> <td>9h</td> <td>Reserved</td> </tr> <tr> <td>Ah</td> <td>10 clocks</td> </tr> <tr> <td>Bh</td> <td>Reserved</td> </tr> <tr> <td>Ch</td> <td>12 clocks</td> </tr> <tr> <td>Dh</td> <td>Reserved</td> </tr> <tr> <td>Eh</td> <td>14 clocks</td> </tr> <tr> <td>Fh</td> <td>Reserved</td> </tr> <tr> <td>10h</td> <td>16 clocks</td> </tr> <tr> <td>11h</td> <td>Reserved</td> </tr> <tr> <td>12h</td> <td>18 clocks</td> </tr> <tr> <td>1Fh-13h</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Description	4h-0h	Reserved	8h-5h	8 to 5 clocks	9h	Reserved	Ah	10 clocks	Bh	Reserved	Ch	12 clocks	Dh	Reserved	Eh	14 clocks	Fh	Reserved	10h	16 clocks	11h	Reserved	12h	18 clocks	1Fh-13h	Reserved
Bits	Description																												
4h-0h	Reserved																												
8h-5h	8 to 5 clocks																												
9h	Reserved																												
Ah	10 clocks																												
Bh	Reserved																												
Ch	12 clocks																												
Dh	Reserved																												
Eh	14 clocks																												
Fh	Reserved																												
10h	16 clocks																												
11h	Reserved																												
12h	18 clocks																												
1Fh-13h	Reserved																												

#### **D18F2x[234:230]\_dct[0] DDR3 DRAM Read ODT Pattern [High:Low]**

Reset: 0000\_0000h. See [2.9.3 \[DCT Configuration Registers\]](#). This register is used by BIOS to specify the state of the ODT pins during DDR reads. F2x230 is used to control chip selects 0-3. F2x234 is used to control chip selects 4-7.

See [2.9.7.6.5 \[DRAM ODT Control\]](#) for more information.

Bits	Description
31:28	Reserved.
27:24	<b>RdOdtPatCs73: read ODT pattern chip select [7,3].</b> See: RdOdtPatCs40.
23:20	Reserved.
19:16	<b>RdOdtPatCs62: read ODT pattern chip select [6,2].</b> See: RdOdtPatCs40.
15:12	Reserved.
11:8	<b>RdOdtPatCs51: read ODT pattern chip select [5,1].</b> See: RdOdtPatCs40.
7:4	Reserved.
3:0	<b>RdOdtPatCs40: read ODT pattern chip select [4,0].</b> Read-write. Specifies the state of ODT[3:0] pins when a read occurs to the specified chip select.

#### **D18F2x[23C:238]\_dct[0] DDR3 DRAM Write ODT Pattern [High:Low]**

Reset: 0000\_0000h. See [2.9.3 \[DCT Configuration Registers\]](#). This register is used by BIOS to specify the state of the ODT pins during DDR writes. F2x238 is used to control chip selects 0-3. F2x23C is used to control chip selects 4-7.

See [2.9.7.6.5 \[DRAM ODT Control\]](#) for more information.

Bits	Description
31:28	Reserved.

27:24	<b>WrOdtPatCs73: write ODT pattern chip select [7,3].</b> See: WrOdtPatCs40.
23:20	Reserved.
19:16	<b>WrOdtPatCs62: write ODT pattern chip select [6,2].</b> See: WrOdtPatCs40.
15:12	Reserved.
11:8	<b>WrOdtPatCs51: write ODT pattern chip select [5,1].</b> See: WrOdtPatCs40.
7:4	Reserved.
3:0	<b>WrOdtPatCs40: write ODT pattern chip select [4,0].</b> Read-write. Specifies the state of ODT[3:0] pins when a write occurs to the specified chip select.

### D18F2x240\_dct[0]\_mp[1:0] DDR3 DRAM ODT Control

Reset: 0000\_0000h. See 2.9.3 [DCT Configuration Registers].

Bits	Description								
31:15	Reserved.								
14:12	<b>WrOdtOnDuration: write ODT on duration.</b> Read-write. BIOS: 6. Specifies the number of memory clock cycles that ODT is asserted for writes. <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>5h-0h</td> <td>Reserved</td> </tr> <tr> <td>7h-6h</td> <td>&lt;WrOdtOnDuration&gt; clocks</td> </tr> </tbody> </table>	Bits	Description	5h-0h	Reserved	7h-6h	<WrOdtOnDuration> clocks		
Bits	Description								
5h-0h	Reserved								
7h-6h	<WrOdtOnDuration> clocks								
11	Reserved.								
10:8	<b>WrOdtTrnOnDly: Write ODT Turn On Delay.</b> Read-write. BIOS: 0. Specifies the number of memory clock cycles that ODT assertion is delayed relative to write CAS. <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>0 clocks</td> </tr> <tr> <td>7h-1h</td> <td>&lt;WrOdtTrnOnDly&gt; clocks, Reserved if (WrOdtOnDuration=0)</td> </tr> </tbody> </table>	Bits	Description	0h	0 clocks	7h-1h	<WrOdtTrnOnDly> clocks, Reserved if (WrOdtOnDuration=0)		
Bits	Description								
0h	0 clocks								
7h-1h	<WrOdtTrnOnDly> clocks, Reserved if (WrOdtOnDuration=0)								
7:4	<b>RdOdtOnDuration: Read ODT On Duration.</b> Read-write. BIOS: 6. Specifies the number of memory clock cycles that ODT is asserted for an eight-beat read burst. The controller will shorten the ODT pulse duration by two clock cycles if the burst is chopped. <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>5h-0h</td> <td>Reserved</td> </tr> <tr> <td>9h-6h</td> <td>&lt;RdOdtOnDuration&gt; clocks</td> </tr> <tr> <td>Fh-Ah</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Description	5h-0h	Reserved	9h-6h	<RdOdtOnDuration> clocks	Fh-Ah	Reserved
Bits	Description								
5h-0h	Reserved								
9h-6h	<RdOdtOnDuration> clocks								
Fh-Ah	Reserved								
3:0	<b>RdOdtTrnOnDly: Read ODT Turn On Delay.</b> Read-write. BIOS: MAX(0, D18F2x200_dct[0]_mp[1:0][Tcl] - D18F2x20C_dct[0]_mp[1:0][Tcw1]). Specifies the number of clock cycles that ODT assertion is delayed relative to read CAS. <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>0 clocks</td> </tr> <tr> <td>Fh-0h</td> <td>&lt;RdOdtTrnOnDly&gt; clocks, Reserved if (RdOdtOnDuration=0)</td> </tr> </tbody> </table>	Bits	Description	0h	0 clocks	Fh-0h	<RdOdtTrnOnDly> clocks, Reserved if (RdOdtOnDuration=0)		
Bits	Description								
0h	0 clocks								
Fh-0h	<RdOdtTrnOnDly> clocks, Reserved if (RdOdtOnDuration=0)								

### D18F2x244\_dct[0] DRAM Controller Miscellaneous 3

Reset: 0000\_0000h. See 2.9.3 [DCT Configuration Registers].

Bits	Description
31:8	Reserved.

7:4	Reserved.										
3:0	<p><b>PrtlChPDDynDly: partial channel power down dynamic delay.</b> Read-write. BIOS:4h. Specifies the channel idle hysteresis for fast exit/slow exit mode changes when <a href="#">D18F2xA8_dct[0][PrtlChP-DEnhEn]=1</a>.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>0 clocks</td> </tr> <tr> <td>7h-1h</td> <td>&lt;PrtlChPDDynDly*32&gt; clocks</td> </tr> <tr> <td>8h</td> <td>256 clocks</td> </tr> <tr> <td>Fh-9h</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Description	0h	0 clocks	7h-1h	<PrtlChPDDynDly*32> clocks	8h	256 clocks	Fh-9h	Reserved
Bits	Description										
0h	0 clocks										
7h-1h	<PrtlChPDDynDly*32> clocks										
8h	256 clocks										
Fh-9h	Reserved										

**D18F2x248\_dct[0]\_mp[1:0] DRAM Power Management 0**

Reset: 0000\_0A03h. See [2.9.3 \[DCT Configuration Registers\]](#).

Bits	Description										
31	<p><b>RxChMntClkEn: Receive channel maintenance clocks.</b> Read-Write. BIOS: 0. 1=Enable receive channel maintenance clocks to improve internal timing margin at the cost of some extra power. 0=Disable clocks. To disable clocks, BIOS must first disable clock generation in the phy (see <a href="#">D18F2x9C_x0D0F_0[F,8:0]13_dct[0]_mp[1:0][RxSsbMntClkEn]</a>).</p>										
30	Reserved.										
29:24	<p><b>AggrPDDelay: aggressive power down delay.</b> Read-Write. BIOS: 20h. Specifies a hysteresis count from the last DRAM activity for the DCT to close pages prior to precharge power down. Reserved if <a href="#">D18F2xA8_dct[0][AggrPDEn]==0</a>. See <a href="#">PchgPDEnDelay</a> and <a href="#">D18F2x94_dct[0][PowerDownEn]</a>.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00h</td> <td>64 clocks</td> </tr> <tr> <td>01h</td> <td>1 clock</td> </tr> <tr> <td>3Eh-02h</td> <td>&lt;AggrPDDelay&gt; clocks</td> </tr> <tr> <td>3Fh</td> <td>63 clocks</td> </tr> </tbody> </table>	Bits	Description	00h	64 clocks	01h	1 clock	3Eh-02h	<AggrPDDelay> clocks	3Fh	63 clocks
Bits	Description										
00h	64 clocks										
01h	1 clock										
3Eh-02h	<AggrPDDelay> clocks										
3Fh	63 clocks										
23:22	Reserved.										
21:16	<p><b>PchgPDEnDelay: precharge power down entry delay.</b> Read-write. BIOS: IF (<a href="#">D18F2xA8_dct[0][AggrPDEn]</a>) THEN (<a href="#">D18F2x200_dct[0]_mp[1:0][Tcl]</a> + 5 + CEIL((MAX(<a href="#">D18F2x9C_x0000_00[2A:10]_dct[0]_mp[1:0][DqsRcvEnGrossDelay]</a>) + 0.5) / 2)) ELSE 00h ENDIF. Specifies the power down entry delay. If <a href="#">D18F2xA8_dct[0][AggrPDEn]==0</a>, this delay behaves as a hysteresis. This field must satisfy the minimum power down entry delay requirements. See also <a href="#">D18F2x94_dct[0][PowerDownEn]</a>. <math>PchgPDEnDelay == 0 \parallel PchgPDEnDelay \geq D18F2x200\_dct[0]_mp[1:0][Tcl] + 5 + CEIL((MAX(D18F2x9C\_x0000\_00[2A:10]_dct[0]_mp[1:0][DqsRcvEnGrossDelay]) + 0.5) / 2)</math>.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00h</td> <td>64 clocks</td> </tr> <tr> <td>01h</td> <td>1 clock</td> </tr> <tr> <td>3Eh-02h</td> <td>&lt;PchgPDEnDelay&gt; clocks</td> </tr> <tr> <td>3Fh</td> <td>63 clocks</td> </tr> </tbody> </table>	Bits	Description	00h	64 clocks	01h	1 clock	3Eh-02h	<PchgPDEnDelay> clocks	3Fh	63 clocks
Bits	Description										
00h	64 clocks										
01h	1 clock										
3Eh-02h	<PchgPDEnDelay> clocks										
3Fh	63 clocks										
15:13	Reserved.										

12:8	<b>Txpdll: exit DLL and precharge powerdown to command delay.</b> Read-write. Specifies the minimum time that the DCT waits to issue a command after exiting precharge powerdown mode if the DLL was also disabled. <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>09h-00h</td> <td>Reserved</td> </tr> <tr> <td>1Dh-0Ah</td> <td>&lt;Txpdll&gt; clocks</td> </tr> <tr> <td>1Fh-1Eh</td> <td>Reserved</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Description</u>	09h-00h	Reserved	1Dh-0Ah	<Txpdll> clocks	1Fh-1Eh	Reserved
<u>Bits</u>	<u>Description</u>								
09h-00h	Reserved								
1Dh-0Ah	<Txpdll> clocks								
1Fh-1Eh	Reserved								
7:4	Reserved.								
3:0	<b>Txp: exit precharge PD to command delay.</b> Read-write. Specifies the minimum time that the DCT waits to issue a command after exiting precharge powerdown mode. <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>2h-0h</td> <td>Reserved</td> </tr> <tr> <td>8h-3h</td> <td>&lt;Txp&gt; clocks</td> </tr> <tr> <td>Fh-9h</td> <td>Reserved</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Description</u>	2h-0h	Reserved	8h-3h	<Txp> clocks	Fh-9h	Reserved
<u>Bits</u>	<u>Description</u>								
2h-0h	Reserved								
8h-3h	<Txp> clocks								
Fh-9h	Reserved								

**D18F2x24C\_dct[0] DDR3 DRAM Power Management 1**

Reset: 0214\_0803h. See 2.9.3 [DCT Configuration Registers].

Bits	Description								
31:30	Reserved.								
29:24	<b>Tcksrx: clock stable to self refresh exit delay.</b> Read-write. Specifies the minimum time in memory clock cycles that the DCT waits to assert CKE after clock frequency is stable. <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>01h-00h</td> <td>Reserved</td> </tr> <tr> <td>0Eh-02h</td> <td>&lt;Tcksrx&gt; clocks</td> </tr> <tr> <td>3Fh-0Fh</td> <td>Reserved</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Description</u>	01h-00h	Reserved	0Eh-02h	<Tcksrx> clocks	3Fh-0Fh	Reserved
<u>Bits</u>	<u>Description</u>								
01h-00h	Reserved								
0Eh-02h	<Tcksrx> clocks								
3Fh-0Fh	Reserved								
23:22	Reserved.								
21:16	<b>Tcksre: self refresh to command delay.</b> Read-write. Specifies the minimum time in memory clock cycles that the DCT waits to remove external clocks after entering self refresh or powerdown. <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>04h-00h</td> <td>Reserved</td> </tr> <tr> <td>27h-05h</td> <td>&lt;Tcksre&gt; clocks</td> </tr> <tr> <td>3Fh-28h</td> <td>Reserved</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Description</u>	04h-00h	Reserved	27h-05h	<Tcksre> clocks	3Fh-28h	Reserved
<u>Bits</u>	<u>Description</u>								
04h-00h	Reserved								
27h-05h	<Tcksre> clocks								
3Fh-28h	Reserved								
15:14	Reserved.								
13:8	<b>Tckesr: self refresh to command delay.</b> Read-write. Specifies the minimum time in memory clock cycles that the DCT waits to issue a command after entering self refresh. <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>01h-00h</td> <td>Reserved</td> </tr> <tr> <td>2Bh-02h</td> <td>&lt;Tckesr&gt; clocks</td> </tr> <tr> <td>3Fh-2Ch</td> <td>Reserved</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Description</u>	01h-00h	Reserved	2Bh-02h	<Tckesr> clocks	3Fh-2Ch	Reserved
<u>Bits</u>	<u>Description</u>								
01h-00h	Reserved								
2Bh-02h	<Tckesr> clocks								
3Fh-2Ch	Reserved								

7:4	Reserved.								
3:0	<b>Tpd: minimum power down entry to exit.</b> Read-write. Specifies minimum time in memory clock cycles for powerdown entry to exit timing. <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Reserved</td> </tr> <tr> <td>Ah-1h</td> <td>&lt;Tpd&gt; clocks</td> </tr> <tr> <td>Fh-Bh</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Description	0h	Reserved	Ah-1h	<Tpd> clocks	Fh-Bh	Reserved
Bits	Description								
0h	Reserved								
Ah-1h	<Tpd> clocks								
Fh-Bh	Reserved								

### **D18F2x250\_dct[0] DRAM Loopback and Training Control**

Reset: 0000\_0000h. See 2.9.3 [DCT Configuration Registers]. See 2.9.8 [Continuous Pattern Generation].

Bits	Description														
31:21	Reserved.														
20:18	<b>DataPatGenSel: data pattern generator select.</b> IF (D18F2x118[LockDramCfg]) THEN Read-only. ELSE Read-write. ENDIF. <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>PRBS23 I.</td> </tr> <tr> <td>001b</td> <td>PRBS23 II. Within a byte lane, the DQ[n] value is offset 8n bit times from DQ[0].</td> </tr> <tr> <td>010b</td> <td>Configurable data pattern. See D18F2x2[B4,B0,AC,A8]_dct[0].</td> </tr> <tr> <td>011b</td> <td>Configurable data pattern with circular lane shift. See D18F2x2[B4,B0,AC,A8]_dct[0].</td> </tr> <tr> <td>100b</td> <td>PRBS23 III. Within a byte lane, each DQ[n] value is equal to DQ[0].</td> </tr> <tr> <td>111b-101b</td> <td>Reserved.</td> </tr> </tbody> </table>	Bits	Description	000b	PRBS23 I.	001b	PRBS23 II. Within a byte lane, the DQ[n] value is offset 8n bit times from DQ[0].	010b	Configurable data pattern. See D18F2x2[B4,B0,AC,A8]_dct[0].	011b	Configurable data pattern with circular lane shift. See D18F2x2[B4,B0,AC,A8]_dct[0].	100b	PRBS23 III. Within a byte lane, each DQ[n] value is equal to DQ[0].	111b-101b	Reserved.
Bits	Description														
000b	PRBS23 I.														
001b	PRBS23 II. Within a byte lane, the DQ[n] value is offset 8n bit times from DQ[0].														
010b	Configurable data pattern. See D18F2x2[B4,B0,AC,A8]_dct[0].														
011b	Configurable data pattern with circular lane shift. See D18F2x2[B4,B0,AC,A8]_dct[0].														
100b	PRBS23 III. Within a byte lane, each DQ[n] value is equal to DQ[0].														
111b-101b	Reserved.														
17	<b>ActPchgGenEn: activate precharge generation enable.</b> IF (D18F2x118[LockDramCfg]) THEN Read-only. ELSE Read-write. ENDIF. 1=The DCT generates ACT and PRE traffic in the available command bandwidth from SendCmd. 0=Traffic generation is not enabled. Reserved if ~CmdTestEnable.														
16	Reserved.														
15	Reserved.														
14	Reserved.														
13	<b>LfsrRollOver: LFSR roll over.</b> IF (D18F2x118[LockDramCfg]) THEN Read-only. ELSE Read-write. ENDIF. Specifies the behavior of DataPrbsSeed and the data comparison logic if the generated address wraps around to equal D18F2x25[8,4]_dct[0][TgtAddress]. 0=The PRBS will not be re-seeded. 1=The PRBS will be re-seeded.														
12	<b>CmdSendInProg: command in progress.</b> Read-only; updated-by-hardware. 0=DCT is idle. 1=DCT is busy.														
11	<b>SendCmd: send command.</b> IF (D18F2x118[LockDramCfg]) THEN Read-only. ELSE Read-write. ENDIF. 0=Stop command generation. 1=Begin command generation as specified in CmdTgt, CmdType, and D18F2x260_dct[0][CmdCount]. BIOS must set this field to a 0 after a command series is completed. Reserved if ~CmdTestEnable.														
10	<b>TestStatus: test status.</b> Read-only. 0=Command generation is in progress. 1=Command generation has completed. Reserved if ~(SendCmd & (D18F2x260_dct[0][CmdCount] > 0   StopOnErr)).														

9:8	<p><b>CmdTgt: command target.</b> IF (D18F2x118[LockDramCfg]) THEN Read-only. ELSE Read-write. ENDIF. Specifies the SendCmd command target address mode. See D18F2x25[8,4]_dct[0].</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Issue commands to address Target A</td> </tr> <tr> <td>01b</td> <td>Issue alternating commands to address Target A and Target B</td> </tr> <tr> <td>11b-10b</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Description	00b	Issue commands to address Target A	01b	Issue alternating commands to address Target A and Target B	11b-10b	Reserved		
Bits	Description										
00b	Issue commands to address Target A										
01b	Issue alternating commands to address Target A and Target B										
11b-10b	Reserved										
7:5	<p><b>CmdType: command type.</b> IF (D18F2x118[LockDramCfg]) THEN Read-only. ELSE Read-write. ENDIF. Specifies the SendCmd command type.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>Read</td> </tr> <tr> <td>001b</td> <td>Write</td> </tr> <tr> <td>010b</td> <td>Alternating write and read</td> </tr> <tr> <td>111b-011b</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Description	000b	Read	001b	Write	010b	Alternating write and read	111b-011b	Reserved
Bits	Description										
000b	Read										
001b	Write										
010b	Alternating write and read										
111b-011b	Reserved										
4	<p><b>StopOnErr: stop on error.</b> IF (D18F2x118[LockDramCfg]) THEN Read-only. ELSE Read-write. ENDIF. Specifies the DCT behavior if a data comparison error occurs. 1=Stop command generation. 0=Continue command generation. If StopOnErr=1, BIOS must program ResetAllErr=1 when programming SendCmd=1.</p>										
3	<p><b>ResetAllErr: reset all errors.</b> IF (D18F2x118[LockDramCfg]) THEN Read-only. ELSE Read; write-1-only; cleared-by-hardware. ENDIF. 1=Clear error status bits and error counters in D18F2x264_dct[0], D18F2x268_dct[0], and D18F2x26C_dct[0].</p>										
2	<p><b>CmdTestEnable: command test enable.</b> IF (D18F2x118[LockDramCfg]) THEN Read-only. ELSE Read-write. ENDIF. 0=Disable the command generation mode. 1=Enable the command generation mode. See SendCmd.</p>										
1:0	Reserved.										

**D18F2x25[8,4]\_dct[0] DRAM Target [B, A] Base**

Reset: 0000\_0000h. See 2.9.3 [DCT Configuration Registers]. See 2.9.8 [Continuous Pattern Generation].

Table 169: Register Mapping for D18F2x25[8,4]\_dct[0]

Register	Function
D18F2x254_dct[0]	Target A
D18F2x258_dct[0]	Target B

Bits	Description				
31:27	Reserved.				
26:24	<p><b>TgtChipSelect: target chip select.</b> Read-write. Specifies the chip select.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>111b-000b</td> <td>CS&lt;TgtChipSelect&gt;</td> </tr> </tbody> </table>	Bits	Description	111b-000b	CS<TgtChipSelect>
Bits	Description				
111b-000b	CS<TgtChipSelect>				
23:21	<b>TgtBank: target bank [2:0].</b> Read-write. Specifies the bank address.				
20:16	Reserved.				

15:10	<b>TgtAddress[15:10]: target address [15:10].</b> Read-write. Specifies the upper column address bits [15:10]. Software must always program bit 10 and bit 12 equal to 0.
9:0	<b>TgtAddress[9:0]: target address [9:0].</b> Read-write. Specifies the column address bits [9:0]. The address sequencing in a command series occurs as follows: TgtAddress[9:3] is incremented by one with wrap around. The increment occurs after each command if D18F2x250_dct[0][CmdType] = 00xb or if (D18F2x250_dct[0][CmdType] = 010b and D18F2x250_dct[0][CmdTgt] = 01b). The increment occurs after each command pair if (D18F2x250_dct[0][CmdType] = 010b and D18F2x250_dct[0][CmdTgt] = 00b).

**D18F2x25C\_dct[0] DRAM Command 0**

Reset: 0000\_0001h. See 2.9.3 [DCT Configuration Registers]. See 2.9.8 [Continuous Pattern Generation].

Bits	Description										
31:22	<b>BubbleCnt2: bubble count 2.</b> See: BubbleCnt. Specifies the number of NOP commands inserted after the last clock of virtual CAS of each read-burst operation in alternating write and read mode. Defined only if (D18F2x250_dct[0][CmdType] == 010b); otherwise reserved.										
21:12	<b>BubbleCnt: bubble count.</b> Read-write. Specifies the number of NOP commands inserted after the last clock of virtual CAS of the last command of the command stream specified by CmdStreamLen. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>000h</td> <td>0 command bubbles</td> </tr> <tr> <td>3FEh-001h</td> <td>&lt;BubbleCnt&gt; command bubbles</td> </tr> <tr> <td>3FFh</td> <td>3FFh command bubbles</td> </tr> </tbody> </table>	Bits	Description	000h	0 command bubbles	3FEh-001h	<BubbleCnt> command bubbles	3FFh	3FFh command bubbles		
Bits	Description										
000h	0 command bubbles										
3FEh-001h	<BubbleCnt> command bubbles										
3FFh	3FFh command bubbles										
11:9	Reserved.										
8	<b>CmdTimingEn: command timing enable.</b> Read-write. 1=Forces DCT to schedule commands initiated by D18F2x250_dct[0][SendCmd] to adhere to the same DRAM timing parameters as normal traffic. 0=Commands initiated by D18F2x250_dct[0][SendCmd] ignore DRAM timing parameters.										
7:0	<b>CmdStreamLen: command stream length.</b> Read-write. Specifies the number of commands generated before BubbleCnt bubbles are inserted. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00h</td> <td>Reserved</td> </tr> <tr> <td>01h</td> <td>1 command</td> </tr> <tr> <td>FEh-02h</td> <td>&lt;CmdStreamLen&gt; commands; defined only if ~(D18F2x250_dct[0][CmdType]=010b)</td> </tr> <tr> <td>FFh</td> <td>255 commands; defined only if ~(D18F2x250_dct[0][CmdType]=010b)</td> </tr> </tbody> </table>	Bits	Description	00h	Reserved	01h	1 command	FEh-02h	<CmdStreamLen> commands; defined only if ~(D18F2x250_dct[0][CmdType]=010b)	FFh	255 commands; defined only if ~(D18F2x250_dct[0][CmdType]=010b)
Bits	Description										
00h	Reserved										
01h	1 command										
FEh-02h	<CmdStreamLen> commands; defined only if ~(D18F2x250_dct[0][CmdType]=010b)										
FFh	255 commands; defined only if ~(D18F2x250_dct[0][CmdType]=010b)										

**D18F2x260\_dct[0] DRAM Command 1**

Reset: 0000\_0000h. See 2.9.3 [DCT Configuration Registers]. See 2.9.8 [Continuous Pattern Generation].

Bits	Description
------	-------------

31:21	Reserved.						
20:0	<p><b>CmdCount: command count.</b> Read-write. Specifies the maximum number of commands to generate when <code>D18F2x250_dct[0][SendCmd]=1</code>. See also <code>D18F2x250_dct[0][StopOnErr]</code>.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Infinite commands.</td> </tr> <tr> <td>1F_FFFFh-1h</td> <td>&lt;CmdCount&gt; commands</td> </tr> </tbody> </table>	Bits	Description	0h	Infinite commands.	1F_FFFFh-1h	<CmdCount> commands
Bits	Description						
0h	Infinite commands.						
1F_FFFFh-1h	<CmdCount> commands						

### D18F2x264\_dct[0] DRAM Status 0

Reset: 0000\_0000h. See 2.9.3 [DCT Configuration Registers]. See 2.9.8 [Continuous Pattern Generation].

Bits	Description										
31:25	<p><b>ErrDqNum: error DQ number.</b> Read-only. Indicates the DQ bit of the first error occurrence when <code>D18F2x264_dct[0][ErrCnt] &gt; 0</code>. Cleared by <code>D18F2x250_dct[0][ResetAllErr]</code>.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>3Fh-00h</td> <td>Data[&lt;ErrDqNum&gt;]</td> </tr> <tr> <td>47h-40h</td> <td>ECC[7:0]</td> </tr> <tr> <td>7Fh-48h</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Description	3Fh-00h	Data[<ErrDqNum>]	47h-40h	ECC[7:0]	7Fh-48h	Reserved		
Bits	Description										
3Fh-00h	Data[<ErrDqNum>]										
47h-40h	ECC[7:0]										
7Fh-48h	Reserved										
24:0	<p><b>ErrCnt: error count.</b> Read; set-by-hardware; write-1-to-clear. Specifies a saturating counter indicating the number of DQ bit errors detected. Counts a maximum of 1 error per bit-lane per each bit-time. Status is accumulated until cleared by <code>D18F2x250_dct[0][ResetAllErr]</code>. Errors can be masked on per-bit basis by programming <code>D18F2x274_dct[0]</code>, <code>D18F2x278_dct[0]</code>, and <code>D18F2x27C_dct[0]</code>.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>0 errors</td> </tr> <tr> <td>1FF_FFFDh-1h</td> <td>&lt;ErrCnt&gt; errors</td> </tr> <tr> <td>1FF_FFFEh</td> <td>1FF_FFFEh errors</td> </tr> <tr> <td>1FF_FFFFh</td> <td>1FF_FFFFh or more errors</td> </tr> </tbody> </table>	Bits	Description	0h	0 errors	1FF_FFFDh-1h	<ErrCnt> errors	1FF_FFFEh	1FF_FFFEh errors	1FF_FFFFh	1FF_FFFFh or more errors
Bits	Description										
0h	0 errors										
1FF_FFFDh-1h	<ErrCnt> errors										
1FF_FFFEh	1FF_FFFEh errors										
1FF_FFFFh	1FF_FFFFh or more errors										

### D18F2x268\_dct[0] DRAM Status 1

Reset: 0000\_0000h. See 2.9.3 [DCT Configuration Registers]. See 2.9.8 [Continuous Pattern Generation].

Bits	Description										
31:20	Reserved.										
19:0	<p><b>NibbleErrSts: nibble error status.</b> Read-only. Indicates error detection status on a per nibble basis when <code>D18F2x264_dct[0][ErrCnt] &gt; 0</code>. Status is accumulated until cleared by <code>D18F2x250_dct[0][ResetAllErr]</code>.</p> <table border="1"> <thead> <tr> <th>Bit</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>[15:0]</td> <td>Data[&lt;(NibbleErrSts*4)+3&gt;:&lt;NibbleErrSts*4&gt;]</td> </tr> <tr> <td>[16]</td> <td>ECC[3:0]</td> </tr> <tr> <td>[17]</td> <td>ECC[7:4]</td> </tr> <tr> <td>[19:18]</td> <td>Reserved</td> </tr> </tbody> </table>	Bit	Description	[15:0]	Data[<(NibbleErrSts*4)+3>:<NibbleErrSts*4>]	[16]	ECC[3:0]	[17]	ECC[7:4]	[19:18]	Reserved
Bit	Description										
[15:0]	Data[<(NibbleErrSts*4)+3>:<NibbleErrSts*4>]										
[16]	ECC[3:0]										
[17]	ECC[7:4]										
[19:18]	Reserved										

### D18F2x26C\_dct[0] DRAM Status 2

Reset: 0000\_0000h. See 2.9.3 [DCT Configuration Registers]. See 2.9.8 [Continuous Pattern Generation].



Bits	Description								
31:18	Reserved.								
17:0	<b>NibbleErr180Sts: nibble error 180 status.</b> Read-only. Indicates error detection status on a per nibble basis when <code>D18F2x264_dct[0][ErrCnt] &gt; 0</code> , comparing read data against data shifted 1-bit time earlier. Status is accumulated until cleared by <code>D18F2x250_dct[0][ResetAllErr]</code> . <table border="1"> <thead> <tr> <th>Bit</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>[15:0]</td> <td>Data[&lt;(NibbleErr180Sts)*4&gt;+3&gt;:&lt;NibbleErr180Sts*4&gt;]</td> </tr> <tr> <td>[16]</td> <td>ECC[3:0]</td> </tr> <tr> <td>[17]</td> <td>ECC[7:4]</td> </tr> </tbody> </table>	Bit	Description	[15:0]	Data[<(NibbleErr180Sts)*4>+3>:<NibbleErr180Sts*4>]	[16]	ECC[3:0]	[17]	ECC[7:4]
Bit	Description								
[15:0]	Data[<(NibbleErr180Sts)*4>+3>:<NibbleErr180Sts*4>]								
[16]	ECC[3:0]								
[17]	ECC[7:4]								

**D18F2x270\_dct[0] DRAM PRBS**

See 2.9.3 [DCT Configuration Registers]. See 2.9.8 [Continuous Pattern Generation].

Bits	Description
31	Reserved.
30:24	Reserved.
23:19	Reserved.
18:0	<b>DataPrbsSeed: data PRBS seed.</b> Read-write. Reset: 7FFFFh. Specifies the seed value used for creating pseudo random traffic on the data bus. This register must be written with a non-zero seed value.

**D18F2x274\_dct[0] DRAM DQ Mask Low**

See `D18F1x10C[DctCfgSel]`. See 2.9.3 [DCT Configuration Registers]. See 2.9.8 [Continuous Pattern Generation].

Bits	Description								
31:0	<b>DQMask[31:0]: DQ mask.</b> Read-write. <code>DQMask[63:0] = {D18F2x278_dct[0][DQMask[63:32]], DQMask[31:0]}</code> . Reset: 0000_0000_0000_0000h. 1=The corresponding DQ bit will not be compared. 0=The corresponding DQ bit will be compared. See <code>D18F2x264_dct[0][ErrCnt]</code> . <table border="1"> <thead> <tr> <th>Bit</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>[0]</td> <td>Data[0]</td> </tr> <tr> <td>[62:1]</td> <td>Data[&lt;DQMask&gt;]</td> </tr> <tr> <td>[63]</td> <td>Data[63]</td> </tr> </tbody> </table>	Bit	Description	[0]	Data[0]	[62:1]	Data[<DQMask>]	[63]	Data[63]
Bit	Description								
[0]	Data[0]								
[62:1]	Data[<DQMask>]								
[63]	Data[63]								

**D18F2x278\_dct[0] DRAM DQ Mask High**

Bits	Description
31:0	<b>DQMask[63:32]: DQ mask.</b> See: <code>D18F2x274_dct[0][DQMask[31:0]]</code> .

**D18F2x27C\_dct[0] DRAM ECC and EDC Mask**

Reset: 0000\_0000h. See 2.9.3 [DCT Configuration Registers]. See 2.9.8 [Continuous Pattern Generation].

Bits	Description								
31:20	Reserved.								
19:16	Reserved.								
15:8	Reserved.								
7:0	<b>EccMask: ECC mask.</b> Read-write. 1=The corresponding ECC DQ bit will not be compared. 0=The corresponding ECC DQ bit will be compared. See <a href="#">D18F2x264_dct[0][ErrCnt]</a> .  <table border="1"> <thead> <tr> <th>Bit</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>[0]</td> <td>ECC[0]</td> </tr> <tr> <td>[6:1]</td> <td>ECC[&lt;EccMask&gt;]</td> </tr> <tr> <td>[7]</td> <td>ECC[7].</td> </tr> </tbody> </table>	Bit	Description	[0]	ECC[0]	[6:1]	ECC[<EccMask>]	[7]	ECC[7].
Bit	Description								
[0]	ECC[0]								
[6:1]	ECC[<EccMask>]								
[7]	ECC[7].								

### **D18F2x280\_dct[0] DRAM DQ Pattern Override 0**

See [D18F1x10C\[DctCfgSel\]](#). See [2.9.3 \[DCT Configuration Registers\]](#). See [2.9.8 \[Continuous Pattern Generation\]](#).

Bits	Description								
31:0	<b>DQPatOvrEn[31:0]: DQ pattern override enable.</b> Read-write. DQPatOvrEn[63:0] = { <a href="#">D18F2x284_dct[0][DQPatOvrEn[63:32]]</a> , DQPatOvrEn[31:0]}. Reset: 0000_0000_0000_0000h. 1=The pattern on this DQ bit lane will be overridden with a static value specified by <a href="#">D18F2x288_dct[0][PatOvrVal]</a> . 0=The pattern on this DQ bit lane will not be overridden. Reserved if <a href="#">D18F2x250_dct[0][DataPatGenSel]</a> != 10b.  <table border="1"> <thead> <tr> <th>Bit</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>[0]</td> <td>Data[0]</td> </tr> <tr> <td>[62:1]</td> <td>Data[&lt;DQPatOvrEn[31:0]&gt;]</td> </tr> <tr> <td>[63]</td> <td>Data[63]</td> </tr> </tbody> </table>	Bit	Description	[0]	Data[0]	[62:1]	Data[<DQPatOvrEn[31:0]>]	[63]	Data[63]
Bit	Description								
[0]	Data[0]								
[62:1]	Data[<DQPatOvrEn[31:0]>]								
[63]	Data[63]								

### **D18F2x284\_dct[0] DRAM DQ Pattern Override 1**

Bits	Description
31:0	<b>DQPatOvrEn[63:32]: DQ pattern override enable.</b> See: <a href="#">D18F2x280_dct[0][DQPatOvrEn[31:0]]</a> .

### **D18F2x288\_dct[0] DRAM DQ Pattern Override 2**

Reset: 0000\_0000h. See [D18F1x10C\[DctCfgSel\]](#). See [2.9.3 \[DCT Configuration Registers\]](#). See [2.9.8 \[Continuous Pattern Generation\]](#).

Bits	Description										
31:24	<p><b>XorPatOvr: xor pattern override.</b> Read-write. Specifies the override data pattern used for creating traffic on the data bus. The output data for each DQ of each byte lane will be XOR'd with the specified values. Reserved if <code>D18F2x250_dct[0][DataPatGenSel] != 100b</code>.</p> <table border="1"> <thead> <tr> <th>Bit</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>[0]</td> <td>DQ[0]</td> </tr> <tr> <td>[1]</td> <td>DQ[1]</td> </tr> <tr> <td>[6:2]</td> <td>DQ[&lt;XorPatOvr&gt;]</td> </tr> <tr> <td>[7]</td> <td>DQ[7]</td> </tr> </tbody> </table>	Bit	Description	[0]	DQ[0]	[1]	DQ[1]	[6:2]	DQ[<XorPatOvr>]	[7]	DQ[7]
Bit	Description										
[0]	DQ[0]										
[1]	DQ[1]										
[6:2]	DQ[<XorPatOvr>]										
[7]	DQ[7]										
23:9	Reserved.										
8	<p><b>PatOvrVal: pattern override value.</b> Read-write. Specifies the override data pattern used for creating traffic on the data bus. 1=Static 1's. 0=Static 0's. Reserved if <code>D18F2x250_dct[0][DataPatGenSel] != 10b</code>.</p>										
7:0	<p><b>EccPatOvrEn: ECC pattern override enable.</b> Read-write. See <code>D18F2x280_dct[0][DQPatOvrEn]</code></p> <table border="1"> <thead> <tr> <th>Bit</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>[0]</td> <td>ECC[0]</td> </tr> <tr> <td>[1]</td> <td>ECC[1]</td> </tr> <tr> <td>[6:2]</td> <td>ECC[&lt;EccPatOvr&gt;]</td> </tr> <tr> <td>[7]</td> <td>ECC[7].</td> </tr> </tbody> </table>	Bit	Description	[0]	ECC[0]	[1]	ECC[1]	[6:2]	ECC[<EccPatOvr>]	[7]	ECC[7].
Bit	Description										
[0]	ECC[0]										
[1]	ECC[1]										
[6:2]	ECC[<EccPatOvr>]										
[7]	ECC[7].										

**D18F2x28C\_dct[0] DRAM Command 2**

Reset: 0000\_0000h. See 2.9.3 [DCT Configuration Registers]. See 2.9.8 [Continuous Pattern Generation]. This register may only be used when `D18F2x250_dct[0][CmdTestEnable]=1`.

Bits	Description				
31	<p><b>SendActCmd: send activate command.</b> Read; write-1-only; cleared-by-hardware. 1=The DCT sends an activate command as specified by ChipSelect, Bank, and Address. This bit is cleared by hardware after the command completes.</p>				
30	<p><b>SendPchgCmd: send precharge all command.</b> Read; write-1-only; cleared-by-hardware. The DCT sends a precharge command based on <code>CmdAddress[10]</code>. This bit is cleared by hardware after the command completes. 0=Command has completed. 1=If (<code>CmdAddress[10]=1</code>) then send a precharge all command as specified by <code>CmdChipSelect</code>; If (<code>CmdAddress[10]=0</code>) then send a precharge command as specified by <code>CmdChipSelect</code>, <code>CmdBank</code>.</p>				
29:22	<p><b>CmdChipSelect: command chip select.</b> Read-write. Specifies the chip select.</p> <table border="1"> <thead> <tr> <th>Bit</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>[7:0]</td> <td>CS&lt;CmdChipSelect&gt;</td> </tr> </tbody> </table>	Bit	Description	[7:0]	CS<CmdChipSelect>
Bit	Description				
[7:0]	CS<CmdChipSelect>				
21:19	<p><b>CmdBank: command bank [2:0].</b> Read-write. Specifies the bank address.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>7h-0h</td> <td>Bank&lt;CmdBank&gt;</td> </tr> </tbody> </table>	Bits	Description	7h-0h	Bank<CmdBank>
Bits	Description				
7h-0h	Bank<CmdBank>				
18	Reserved.				
17:0	<p><b>CmdAddress: command address [17:0].</b> Read-write. Specifies the row address.</p>				

**D18F2x290\_dct[0] DRAM Status 3**

Reset: 0000\_0000h. See 2.9.3 [DCT Configuration Registers]. See 2.9.8 [Continuous Pattern Generation].

Bits	Description				
31:27	Reserved.				
26:24	<b>ErrBeatNum: error beat number.</b> Read-only. Indicates the data beat of the first error occurrence in the command reported by ErrCmdNum when <a href="#">D18F2x264_dct[0][ErrCnt]</a> > 0 and <a href="#">D18F2x260_dct[0][CmdCount]</a> > 0. Cleared by <a href="#">D18F2x250_dct[0][ResetAllErr]</a> . <table border="0"> <tr> <td><u>Bits</u></td> <td><u>Description</u></td> </tr> <tr> <td>7h-0h</td> <td>&lt;ErrBeatNum&gt; beat</td> </tr> </table>	<u>Bits</u>	<u>Description</u>	7h-0h	<ErrBeatNum> beat
<u>Bits</u>	<u>Description</u>				
7h-0h	<ErrBeatNum> beat				
23:21	Reserved.				
20:0	<b>ErrCmdNum: error command number.</b> Read-only. Indicates the command number of the first error occurrence when <a href="#">D18F2x264_dct[0][ErrCnt]</a> > 0 and <a href="#">D18F2x260_dct[0][CmdCount]</a> > 0. Cleared by <a href="#">D18F2x250_dct[0][ResetAllErr]</a> . <table border="0"> <tr> <td><u>Bits</u></td> <td><u>Description</u></td> </tr> <tr> <td>1F_FFFFh-0h</td> <td>&lt;ErrCmdNum&gt; command</td> </tr> </table>	<u>Bits</u>	<u>Description</u>	1F_FFFFh-0h	<ErrCmdNum> command
<u>Bits</u>	<u>Description</u>				
1F_FFFFh-0h	<ErrCmdNum> command				

**D18F2x294\_dct[0] DRAM Status 4**

See 2.9.3 [DCT Configuration Registers]. See 2.9.8 [Continuous Pattern Generation].

Bits	Description								
31:0	<b>DQErr[31:0]: DQ error.</b> Read-only. $DQErr[63:0] = \{D18F2x298\_dct[0][DQErr[63:32]], DQErr[31:0]\}$ . Reset: 0000_0000_0000_0000h. Indicates error detection status on a per bit basis when <a href="#">D18F2x264_dct[0][ErrCnt]</a> > 0. Status is accumulated until cleared by <a href="#">D18F2x250_dct[0][ResetAllErr]</a> . <table border="0"> <tr> <td><u>Bit</u></td> <td><u>Description</u></td> </tr> <tr> <td>[0]</td> <td>Data[0]</td> </tr> <tr> <td>[62:1]</td> <td>Data[&lt;DQErr&gt;]</td> </tr> <tr> <td>[63]</td> <td>Data[63]</td> </tr> </table>	<u>Bit</u>	<u>Description</u>	[0]	Data[0]	[62:1]	Data[<DQErr>]	[63]	Data[63]
<u>Bit</u>	<u>Description</u>								
[0]	Data[0]								
[62:1]	Data[<DQErr>]								
[63]	Data[63]								

**D18F2x298\_dct[0] DRAM Status 5**

Bits	Description
31:0	<b>DQErr[63:32]: DQ error.</b> See: <a href="#">D18F2x294_dct[0][DQErr[31:0]]</a> .

**D18F2x29C\_dct[0] DRAM Status 6**

Reset: 0000\_0000h. See 2.9.3 [DCT Configuration Registers]. See 2.9.8 [Continuous Pattern Generation].

Bits	Description
31:20	Reserved.
19:16	Reserved.

15:8	Reserved.								
7:0	<p><b>EccErr: ECC error.</b> Read-only. Indicates ECC error detection status on a per bit basis when <code>D18F2x264_dct[0][ErrCnt] &gt; 0</code>. Status is accumulated until cleared by <code>D18F2x250_dct[0][ResetAll-Err]</code>.</p> <table border="1"> <thead> <tr> <th>Bit</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>[0]</td> <td>ECC[0]</td> </tr> <tr> <td>[6:1]</td> <td>ECC[&lt;EccErr&gt;]</td> </tr> <tr> <td>[7]</td> <td>ECC[7]</td> </tr> </tbody> </table>	Bit	Description	[0]	ECC[0]	[6:1]	ECC[<EccErr>]	[7]	ECC[7]
Bit	Description								
[0]	ECC[0]								
[6:1]	ECC[<EccErr>]								
[7]	ECC[7]								

**D18F2x2[B4,B0,AC,A8]\_dct[0] DRAM User Data Pattern**

Reset: 0000\_0000h. See 2.9.3 [DCT Configuration Registers]. See 2.9.8 [Continuous Pattern Generation].

Table 170: Register Mapping for D18F2x2[B4,B0,AC,A8]\_dct[0]

Register	Function
D18F2x2A8_dct[0]	Nibble Lanes 0, 4, 8, 12, 16
D18F2x2AC_dct[0]	Nibble Lanes 1, 5, 9, 13, 17
D18F2x2B0_dct[0]	Nibble Lanes 2, 6, 10, 14
D18F2x2B4_dct[0]	Nibble Lanes 3, 7, 11, 15

Table 171: Field Mapping for D18F2x2[B4,B0,AC,A8]\_dct[0]

Register	Bits			
	31:24	23:16	15:8	7:0
D18F2x2A8_dct[0]	DQ3, DQ19, DQ35, DQ51, ECC3	DQ2, DQ18, DQ34, DQ50, ECC2	DQ1, DQ17, DQ33, DQ49, ECC1	DQ0, DQ16, DQ32, DQ48, ECC0
D18F2x2AC_dct[0]	DQ7, DQ23, DQ39, DQ55, ECC7	DQ6, DQ22, DQ38, DQ54, ECC6	DQ5, DQ21, DQ37, DQ53, ECC5	DQ4, DQ20, DQ36, DQ52, ECC4
D18F2x2B0_dct[0]	DQ11, DQ27, DQ43, DQ59	DQ10, DQ26, DQ42, DQ58	DQ9, DQ25, DQ41, DQ57	DQ8, DQ24, DQ40, DQ56
D18F2x2B4_dct[0]	DQ15, DQ31, DQ47, DQ63	DQ14, DQ30, DQ46, DQ62	DQ13, DQ29, DQ45, DQ61	DQ12, DQ28, DQ44, DQ60

Bits	Description
31:24	<b>DataPattern: data pattern.</b> See: <code>D18F2x2[B4,B0,AC,A8]_dct[0][7:0]</code> .

23:16	<b>DataPattern: data pattern.</b> See: <a href="#">D18F2x2[B4,B0,AC,A8]_dct[0][7:0]</a> .								
15:8	<b>DataPattern: data pattern.</b> See: <a href="#">D18F2x2[B4,B0,AC,A8]_dct[0][7:0]</a> .								
7:0	<b>DataPattern: data pattern.</b> Read-write. Specifies a data pattern used for creating traffic on the data bus for the specified bit lanes. See <a href="#">D18F2x250_dct[0][DataPatGenSel]</a> . <table border="1"> <thead> <tr> <th>Bit</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>[0]</td> <td>Pattern Data Beat 0</td> </tr> <tr> <td>[6:1]</td> <td>Pattern Data Beat 1 to Pattern Data Beat 6</td> </tr> <tr> <td>[7]</td> <td>Pattern Data Beat 7</td> </tr> </tbody> </table>	Bit	Description	[0]	Pattern Data Beat 0	[6:1]	Pattern Data Beat 1 to Pattern Data Beat 6	[7]	Pattern Data Beat 7
Bit	Description								
[0]	Pattern Data Beat 0								
[6:1]	Pattern Data Beat 1 to Pattern Data Beat 6								
[7]	Pattern Data Beat 7								

**D18F2x2B8\_dct[0] DRAM Command 3**

Reset: 0000\_0000h. See [2.9.3 \[DCT Configuration Registers\]](#). See [2.9.8 \[Continuous Pattern Generation\]](#).

Bits	Description								
31:16	<b>ActPchgSeq: activate precharge sequence.</b> Read-write. Specifies the command that will be generated in the sequence, corresponding to the sixteen target entries in <a href="#">D18F2x2[C0,BC]_dct[0]</a> . 1=Activate. 0=Precharge. <table border="1"> <thead> <tr> <th>Bit</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>[0]</td> <td>ACT/PRE [0]</td> </tr> <tr> <td>[14:1]</td> <td>ACT/PRE [&lt;ActPchgSeq&gt;]</td> </tr> <tr> <td>[15]</td> <td>ACT/PRE [15]</td> </tr> </tbody> </table>	Bit	Description	[0]	ACT/PRE [0]	[14:1]	ACT/PRE [<ActPchgSeq>]	[15]	ACT/PRE [15]
Bit	Description								
[0]	ACT/PRE [0]								
[14:1]	ACT/PRE [<ActPchgSeq>]								
[15]	ACT/PRE [15]								
15:12	Reserved.								
11:8	<b>ActPchgCmdMin: activate precharge command minimum.</b> Read-write. Specifies the minimum time in memory clock cycles from an activate or precharge command to the next activate or precharge command in the ActPchgSeq sequence. <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Reserved</td> </tr> <tr> <td>Fh-1h</td> <td>[&lt;ActPchgCmdMin&gt;] clocks</td> </tr> </tbody> </table>	Bits	Description	0	Reserved	Fh-1h	[<ActPchgCmdMin>] clocks		
Bits	Description								
0	Reserved								
Fh-1h	[<ActPchgCmdMin>] clocks								
7:0	Reserved.								

**D18F2x2[C0,BC]\_dct[0] DRAM Command 4 & 5**

Reset: 0000\_0000h. See [2.9.3 \[DCT Configuration Registers\]](#). See [2.9.8 \[Continuous Pattern Generation\]](#).

Table 172: Field Mappings for [D18F2x2\[C0,BC\]\\_dct\[0\]](#)

Register	Bits							
	31:28	27:24	23:20	19:16	15:12	11:8	7:4	3:0
<a href="#">D18F2x2BC_dct[0]</a>	ACT/PRE 7	ACT/PRE 6	ACT/PRE 5	ACT/PRE 4	ACT/PRE 3	ACT/PRE 2	ACT/PRE 1	ACT/PRE 0
<a href="#">D18F2x2C0_dct[0]</a>	ACT/PRE 15	ACT/PRE 14	ACT/PRE 13	ACT/PRE 12	ACT/PRE 11	ACT/PRE 10	ACT/PRE 9	ACT/PRE 8

Bits	Description
31:28	<b>ActPchgTgtBank: activate precharge target bank.</b> See: <a href="#">D18F2x2[C0,BC]_dct[0][3:0]</a> .
27:24	<b>ActPchgTgtBank: activate precharge target bank.</b> See: <a href="#">D18F2x2[C0,BC]_dct[0][3:0]</a> .
23:20	<b>ActPchgTgtBank: activate precharge target bank.</b> See: <a href="#">D18F2x2[C0,BC]_dct[0][3:0]</a> .

19:16	<b>ActPchgTgtBank: activate precharge target bank.</b> See: D18F2x2[C0,BC]_dct[0][3:0].						
15:12	<b>ActPchgTgtBank: activate precharge target bank.</b> See: D18F2x2[C0,BC]_dct[0][3:0].						
11:8	<b>ActPchgTgtBank: activate precharge target bank.</b> See: D18F2x2[C0,BC]_dct[0][3:0].						
7:4	<b>ActPchgTgtBank: activate precharge target bank.</b> See: D18F2x2[C0,BC]_dct[0][3:0].						
3:0	<p><b>ActPchgTgtBank: activate precharge target bank.</b> Read-write. Specifies the bank address used for this command of the sequence.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>7h-0h</td> <td>Bank [&lt;ActPchgTgtBank&gt;]</td> </tr> <tr> <td>Fh-8h</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Description	7h-0h	Bank [<ActPchgTgtBank>]	Fh-8h	Reserved
Bits	Description						
7h-0h	Bank [<ActPchgTgtBank>]						
Fh-8h	Reserved						

### D18F2x2E0\_dct[0] Memory P-state Control and Status

See 2.9.3 [DCT Configuration Registers].

Bits	Description								
31	Reserved.								
30	<b>FastMstateDis: fast M-state change disable.</b> Read-write. Reset: 0. 1=The DCT changes MEMCLK frequency only after the NCLK frequency has changed. 0=The DCT changes MEMCLK frequency while the northbridge changes NCLK.								
29	Reserved.								
28:24	<b>M1MemClkFreq: M1 memory clock frequency.</b> Read-write. Reset: 00h. Specifies the frequency of the DRAM interface (MEMCLK) for memory P-state 1. See Table 125 [Valid Values for Memory Clock Frequency Value Definition]. The hardware enforces D18F5x84[DdrMaxRateEnf] when writes to this field occur. See D18F5x84[DdrMaxRate] and D18F5x84[DdrMaxRateEnf]. BIOS must also program D18F2x9C_x0D0F_E000_dct[0]_mp[1:0][Rate] for M1.								
23	Reserved.								
22:20	<p><b>MxMrsEn: Mx Mrs enable.</b> Read-write. Reset:0h. 1=The DCT writes to the DRAM MR after a memory P-state change. 0=The DCT does not write to the DRAM MR.</p> <table border="1"> <thead> <tr> <th>Bit</th> <th>Description, MR value</th> </tr> </thead> <tbody> <tr> <td>[0]</td> <td>MR0, D18F2x2E8_dct[0]_mp[1:0][MxMr0]</td> </tr> <tr> <td>[1]</td> <td>MR1, D18F2x2E8_dct[0]_mp[1:0][MxMr1]</td> </tr> <tr> <td>[2]</td> <td>MR2, D18F2x2EC_dct[0]_mp[1:0][MxMr2]</td> </tr> </tbody> </table>	Bit	Description, MR value	[0]	MR0, D18F2x2E8_dct[0]_mp[1:0][MxMr0]	[1]	MR1, D18F2x2E8_dct[0]_mp[1:0][MxMr1]	[2]	MR2, D18F2x2EC_dct[0]_mp[1:0][MxMr2]
Bit	Description, MR value								
[0]	MR0, D18F2x2E8_dct[0]_mp[1:0][MxMr0]								
[1]	MR1, D18F2x2E8_dct[0]_mp[1:0][MxMr1]								
[2]	MR2, D18F2x2EC_dct[0]_mp[1:0][MxMr2]								
19:1	Reserved.								
0	<b>CurMemPstate: current memory P-state.</b> Reset: 0h. Read-only; updated-by-hardware. Specifies the current memory P-state. 0=M0. 1=M1.								

### D18F2x2E8\_dct[0]\_mp[1:0] MRS Buffer

See 2.9.3 [DCT Configuration Registers].

Bits	Description
------	-------------

31:16	<b>MxMr1: Mx MR1.</b> Read-write. Reset: 0000h. Specifies the value written to DRAM MR1 after a memory P-state change. If the M1 value is the same as the M0 value, then BIOS should optimize P-state switching latency by programming <code>D18F2x2E0_dct[0][MxMrsEn]=0</code> .
15:0	<b>MxMr0: Mx MR0.</b> Read-write. Reset: 0000h. Specifies the value written to DRAM MR0 after a memory P-state change. If the M1 value is the same as the M0 value, then BIOS should optimize P-state switching latency by programming <code>D18F2x2E0_dct[0][MxMrsEn]=0</code> .

### **D18F2x2EC\_dct[0]\_mp[1:0] MRS Buffer**

See 2.9.3 [DCT Configuration Registers].

Bits	Description
31:16	Reserved.
15:0	<b>MxMr2: Mx MR2.</b> Read-write. Reset: 0000h. Specifies the value written to DRAM MR2 after a memory P-state change. If the M1 value is the same as the M0 value, then BIOS should optimize P-state switching latency by programming <code>D18F2x2E0_dct[0][MxMrsEn]=0</code> .

### **D18F2x2F0\_dct[0]\_mp[1:0] DRAM Controller Misc 3**

See 2.9.3 [DCT Configuration Registers].

Bits	Description
31:1	Reserved.
0	<b>EffArbDis: Efficient arbitration disable.</b> Read-write. Reset: 0. BIOS: 0. 0=The DCT optimizes the arbitration phases to improve performance under certain traffic conditions whenever the NCLK to MEMCLK ratio is less than 2:1. 1=The DCT arbitrates normally, at all NCLK:MEMCLK ratios.

### **D18F2x400\_dct[0] GMC to DCT Control 0**

Reset: 0000\_0000h. See 2.9.3 [DCT Configuration Registers].

The GMC to DCT interface controls how DRAM bus resources are allocated and arbitrated between the MCT and the GMC. A token is the unit of available resource and is equivalent to a DCQ entry. A minimum count ensures a number of available DCQ entries. A token limit for MCT or GMC ensures resources are not all allocated to the GMC, or MCT respectively. Limits are configured bimodal: for normal GMC traffic and for when urgent (nominally display refresh) GMC traffic is occurring.

`D18F2x400_dct[0][MctTokenLimit] + D18F2x400_dct[0][GmcTokenLimit] <= 16`.

`D18F2x400_dct[0][GmcTokenLimit] >= 1`.

`D18F2x400_dct[0][MctTokenLimit] >= 1`.

`D18F2x404_dct[0][UrGmcTokenLimit] + D18F2x404_dct[0][UrMctTokenLimit] <= 16`.

`D18F2x404_dct[0][UrGmcMinTokens] + D18F2x404_dct[0][UrMctMinTokens] <= 16`.

`D18F2x400_dct[0][MctTokenLimit] == D18F2x404_dct[0][UrMctTokenLimit]`.

Bits	Description
31:16	Reserved.
15:12	Reserved.



11:8	<b>GmcTokenLimit: GMC token limit.</b> Read-write. BIOS: 4h. Limit of outstanding GMC tokens. <u>Bits</u> <u>Description</u> Fh-0h                      <GmcTokenLimit> tokens
7:4	Reserved.
3:0	<b>MctTokenLimit: MCT token limit.</b> Read-write. BIOS: 4h. Limit of outstanding MCT tokens. <u>Bits</u> <u>Description</u> Fh-0h                      <MctTokenLimit> tokens

**D18F2x404\_dct[0] GMC to DCT Control 1**

Reset: 0000\_0000h. See 2.9.3 [DCT Configuration Registers].

Bits	Description
31	<b>UrgentTknDis: Urgent token disable.</b> Read-write. BIOS: 0. 0=When urgent GMC traffic is requested , override the programmed values in <a href="#">D18F2x400_dct[0]</a> and force the token scheme to heavily weight towards graphics by using the programmable token limits in <a href="#">D18F2x404_dct[0]</a> . 1=Token scheme remains at the previously programmed non-urgent token limits in <a href="#">D18F2x400_dct[0]</a> regardless of urgent GMC traffic.
30:28	Reserved.
27:24	<b>UrGmcMinTokens: Display refresh GMC minimum tokens.</b> Read-write. BIOS: 4. Urgent mode minimum number of tokens assigned to the GMC. <u>Bits</u> <u>Description</u> Fh-0h                      <UrGmcMinTokens> tokens
23:21	Reserved.
20:16	<b>UrGmcTokenLimit: Display refresh GMC token limit.</b> Read-write. BIOS: 04h. Urgent mode limit of outstanding GMC tokens. <u>Bits</u> <u>Description</u> 10h-0h                      <UrGmcTokenLimit> tokens 1Fh-11h                      Reserved
15:12	Reserved.
11:8	<b>UrMctMinTokens: Display refresh MCT minimum tokens.</b> Read-write. BIOS: 4. Urgent mode minimum number of tokens assigned to the MCT. <u>Bits</u> <u>Description</u> Fh-0h                      <UrMctMinTokens> tokens
7:5	Reserved.
4:0	<b>UrMctTokenLimit: Display refresh MCT token limit.</b> Read-write. BIOS: 04h. Urgent mode limit of outstanding MCT tokens. <u>Bits</u> <u>Description</u> 10h-0h                      <UrMctTokenLimit> tokens 1Fh-11h                      Reserved

**D18F2x408\_dct[0] GMC to DCT Control 2**

See 2.9.3 [DCT Configuration Registers].

Bits	Description
31:29	Reserved.

28:24	<p><b>CpuElevPrioPeriod: Cpu elevate priority period.</b> Read-write. Reset: 0. BIOS: Ch. Specifies the hysteresis of how often a new MCT read can be elevated to high priority if no other MCT reads currently exist in the DCQ. If CpuElevPrioPeriod==0, MCT will continuously elevate the priority of a new lone MCT read to high. Reserved if CpuElevPrioDis==1. Since this field controls internal timing in the NCLK domain, external bus equivalence is approximate.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00h</td> <td>hysteresis counter disabled</td> </tr> <tr> <td>1Fh-01h</td> <td>&lt;CpuElevPrioPeriod*32&gt; MEMCLKs</td> </tr> </tbody> </table>	Bits	Description	00h	hysteresis counter disabled	1Fh-01h	<CpuElevPrioPeriod*32> MEMCLKs
Bits	Description						
00h	hysteresis counter disabled						
1Fh-01h	<CpuElevPrioPeriod*32> MEMCLKs						
23:3	Reserved.						
2	<p><b>NonP0UrgentTknDis: non-P0 urgent token disable.</b> Read-write. Reset: 0. BIOS: 0. 0=Switch from normal GMC traffic token scheme defined by <a href="#">D18F2x400_dct[0]</a> to urgent GMC traffic token scheme defined by <a href="#">D18F2x404_dct[0]</a> when all processors are not in software P0 state. 1=Use normal GMC traffic token scheme when all processors are not in software P0 state.</p>						
1	<p><b>TokenAllocSelect: Token allocation select.</b> Read-write. Reset: 0. BIOS: 0. 0=When both the MCT and GMC have less than their maximum outstanding tokens, tokens are allocated by alternating between each. 1= When both the MCT and GMC have less than their maximum outstanding tokens, tokens are allocated to whichever has less (DCQ entries + current outstanding).</p>						
0	<p><b>CpuElevPrioDis: Cpu elevate priority disable.</b> Read-write. Reset: 0. BIOS: 0. 1=Reads from MCT arbitrate with GMC traffic normally. 0=Elevate the priority of a new MCT read to high if no other MCT reads currently exist in the DCQ. This can alleviate CPU stalls during very long graphics requests.</p>						

### **D18F2x420\_dct[0] GMC to DCT FIFO Config 1**

See 2.9.3 [DCT Configuration Registers].

Bits	Description
31:12	Reserved.
11:8	Reserved.
7:4	Reserved.
3:0	Reserved.

### 3.12 Device 18h Function 3 Configuration Registers

See 3.1 [Register Descriptions and Mnemonics]. See 2.7 [Configuration Space].

#### D18F3x00 Device/Vendor ID

Bits	Description
31:16	<b>DeviceID: device ID.</b> Read-only. Value: 1533h.
15:0	<b>VendorID: vendor ID.</b> Read-only. Value: 1022h.

#### D18F3x04 Status/Command

Bits	Description
31:16	<b>Status.</b> Read-only. Reset: 0000h, except bit[20]. Bit[20] is set to indicate the existence of a PCI-defined capability block, if one exists.
15:0	<b>Command.</b> Read-only. Reset: 0000h.

#### D18F3x08 Class Code/Revision ID

Bits	Description
31:8	<b>ClassCode.</b> Read-only. Reset: 060000h. Provides the host bridge class code as defined in the PCI specification.
7:0	<b>RevID: revision ID.</b> Read-only. Reset: 00h.

#### D18F3x0C Header Type

Reset: 0080\_0000h.

Bits	Description
31:0	<b>HeaderTypeReg.</b> Read-only. These bits are fixed at their default values. The header type field indicates that there are multiple functions present in this device.

#### D18F3x34 Capability Pointer

Bits	Description
31:8	Reserved.
7:0	<b>CapPtr.</b> Read-only. Value: 00h.

#### D18F3x40 MCA NB Control

Read-write. Reset: 0000\_0000\_0000\_0000h. [MSR0000\\_0410](#)[31:0] is an alias of [D18F3x40](#). See [MSR0000\\_0410](#)[31:0].

Bits	Description
63:32	Unused.
31	<b>McaCpuDatErrEn: L2 complex data error.</b> 1=Enables MCA reporting of CPU data errors sent to the NB.
30	Unused.
29:28	Unused.
27	Unused.
26	<b>NbArrayParEn: northbridge array parity error reporting enable.</b> 1=Enables reporting of parity errors in the NB arrays.
25	<b>UsPwDatErrEn: upstream data error enable.</b> Read-write. 1=Enables MCA reporting of upstream posted writes in which the EP bit is set.
24:18	Unused.
17	<b>CpPktDatEn: completion packet error reporting enable.</b> Read-write. 1=Enables MCA reporting of completion packets with the EP bit set.
16	<b>NbIntProtEn: northbridge internal bus protocol error reporting enable.</b> Read-write. 1=Enables MCA reporting of protocol errors detected on the northbridge internal bus. When possible, this enable should be cleared before initiating a warm reset to avoid logging spurious errors due to RESET_L signal skew.
15:13	Unused.
12	<b>WDTRptEn: watchdog timer error reporting enable.</b> 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 <a href="#">D18F3x44 [MCA NB Configuration]</a> 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	<b>AtomicRMWEn: atomic read-modify-write error reporting enable.</b> 1=Enables MCA reporting of atomic read-modify-write (RMW) commands received from an IO 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	Unused.
9	<b>TgtAbortEn: target abort error reporting enable.</b> 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	<b>MstrAbortEn: master abort error reporting enable.</b> 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	Unused.
5	<b>SyncPktEn: link sync packet error reporting enable.</b> 1=Enables MCA reporting of link-defined sync error packets detected on link. The NB floods its outgoing link with sync packets after detecting a sync packet on the incoming link independent of the state of this bit.

4:2	Unused.
1	<b>UECCEn: uncorrectable ECC error reporting enable.</b> 1=Enables MCA reporting of DDR3 DRAM uncorrectable ECC errors which are detected in the NB. In some cases data may be forwarded to the core prior to checking ECC in which case the check takes place in one of the other error reporting banks.
0	<b>CECCEn: correctable ECC error reporting enable.</b> 1=Enables MCA reporting of DDR3 DRAM correctable ECC errors which are detected in the NB.

### D18F3x44 MCA NB Configuration

See [D18F3x180 \[Extended NB MCA Configuration\]](#). It is expected that all fields of this register are programmed to the same value in all nodes, except for the fields used for link error injection: GenLinkSel, GenSubLinkSel, GenCrcErrByte1, GenCrcErrByte0.

Bits	Description
31	<b>NbMcaLogEn: northbridge MCA log enable.</b> Read-write. Reset: 0. 1=Enables logging (but not reporting) of NB MCA errors even if MCA is not globally enabled.
30	<b>SyncFloodOnDramAdrParErr: sync flood on DRAM address parity error.</b> Read-write. Reset: 0. BIOS: 1. 1=Enable sync flood on detection of a DRAM address parity error.
29	<b>DisMstAbortCpuErrRsp: master abort CPU error response disable.</b> Read-write. Reset: 0. 1=Disables master abort reporting through the CPU MCA error-reporting banks; Suppresses sending of RDE to CPU; Does not log any MCA information in the NB.
28	<b>DisTgtAbortCpuErrRsp: target abort CPU error response disable.</b> Read-write. Reset: 0. 1=Disables target abort reporting through the CPU MCA error-reporting banks; Suppresses sending of RDE to CPU; Does not log any MCA information in the NB.
27	<b>NbMcaToMstCpuEn: machine check errors to master CPU only.</b> Read-write. Reset: 0. BIOS: 1. 1=NB MCA errors in CMP device are only reported to the node base core (NBC), and the NB MCA registers in MSR space ( <a href="#">MSR0000_0410</a> , <a href="#">MSR0000_0411</a> , <a href="#">MSR0000_0412</a> , <a href="#">MSR0000_0413</a> , <a href="#">MSR0000_0408</a> , and <a href="#">MSRC001_0048</a> ) are only accessible from the NBC; reads of these MSRs from other cores return 0's and writes are ignored. This allows machine check handlers running on different cores to avoid coordinating accesses to the NB MCA registers. This field does not affect PCI-defined configuration space accesses to these registers, which are accessible from all cores. See <a href="#">3.1 [Register Descriptions and Mnemonics]</a> for a description of MSR space and <a href="#">3 [Registers]</a> for PCI-defined configuration space. 0=NB MCA errors may be reported to the core that originated the request, if applicable and known, and the NB MCA registers in MSR space are accessible from any core. Note: <ul style="list-style-type: none"> <li>When the CPU which originated the request is known, it is stored in <a href="#">MSR0000_0411[ErrCoreId]</a>, regardless of the setting of NbMcaToMstCpuEn. See <a href="#">Table 207</a> for errors where ErrCoreId is known.</li> <li>If IO originated the request, then the error is reported to the NBC, regardless of the setting of NbMcaToMstCpuEn.</li> </ul>
26	<b>FlagMcaCorrErr: correctable error MCA exception enable.</b> Read-write. Reset: 0. 1=Raise a machine check exception for correctable and deferred machine check errors which are enabled in <a href="#">D18F3x40</a> .

25	<b>DisPciCfgCpuErrRsp: PCI configuration CPU error response disable.</b> 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. 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 <a href="#">D18F3x180[DisPciCfgCpuMstAbortRsp]</a> , which applies only to master aborts.										
24	<b>IoRdDatErrEn: IO read data error log enable.</b> Read-write. Reset: 0. 1=Enables MCA logging and reporting of errors on transactions from IO devices upon detection of a target abort, master abort, or data error condition. 0=Errors on transactions from IO devices are not logged in MCA, although error responses to the requesting IO device may still be generated.										
23	<b>ChipKillEccCap: chip-kill ECC mode.</b> Read-only; updated-by-hardware. Reset: 0. 1=Chipkill ECC mode capable; ECC checking is based on x8 ECC symbols ( <a href="#">D18F3x180[EccSymbolSize]</a> ) and can be used for chipkill. 0=Chipkill ECC mode not capable; ECC checking is based on two interleaved, unganged 64/8-bit data/ECC lines and x4 ECC symbols and cannot be used for chipkill. See <a href="#">2.14.2 [DRAM ECC Considerations]</a> .										
22	<b>DramEccEn: DRAM ECC enable.</b> Read-write. Reset: 0. 1=Enables ECC check/correct mode. This bit must be set in order for ECC checking/correcting by the NB to be enabled. If set, ECC is checked and correctable errors are corrected irrespective of whether machine check ECC reporting is enabled. The hardware only allows values to be programmed into this field which are consistent with the ECC capabilities of the device as specified in <a href="#">D18F3xE8 [Northbridge Capabilities]</a> . Attempts to write values inconsistent with the capabilities results in this field not being updated. This bit does not affect ECC checking in the northbridge arrays.										
21	<b>SyncFloodOnAnyUcErr: sync flood on any UC error.</b> Read-write. Reset: 0. BIOS: 1. 1=Enable sync flood of all links with sync packets on detection of any NB MCA error that is uncorrectable, including northbridge array errors and link protocol errors.										
20	<b>SyncFloodOnWDT: sync flood on watchdog timer error.</b> Read-write. Reset: 0. BIOS: 1. 1=Enable sync flood of all links with sync packets on detection of a watchdog timer error.										
19:18	<b>GenSubLinkSel: sublink select for CRC error generation.</b> Read-write. Reset: 0. Selects the sublink of a link selected by GenLinkSel to be used for CRC error injection through GenCrcErrByte0 and GenCrcErrByte1. When the link is ganged, GenSubLinkSel must be 00b. When the link is unganged, the following values indicate which sublink is selected: <table border="1" style="margin-left: 20px;"> <thead> <tr> <th><u>Bits</u></th> <th><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Sublink 0</td> </tr> <tr> <td>01b</td> <td>Sublink 1</td> </tr> <tr> <td>10b</td> <td>Reserved</td> </tr> <tr> <td>11b</td> <td>Reserved</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Description</u>	00b	Sublink 0	01b	Sublink 1	10b	Reserved	11b	Reserved
<u>Bits</u>	<u>Description</u>										
00b	Sublink 0										
01b	Sublink 1										
10b	Reserved										
11b	Reserved										
17	<b>GenCrcErrByte1: generate CRC error on byte lane 1.</b> Read-Write. Reset: 0. 1=For ganged links (see GenSubLinkSel), a CRC error is injected on byte lane 1 of the link specified by GenLinkSel. For ganged links in retry mode or unganged links, this field is reserved, and GenCrcErrByte0 must be used. The data carried by the link is unaffected. This bit is cleared after the error has been generated.										
16	<b>GenCrcErrByte0: generate CRC error on byte lane 0.</b> Read-Write. Reset: 0. 1=Causes a CRC error to be injected on byte lane 0 of the link specified by GenLinkSel and the sublink specified by GenSubLinkSel. The data carried by the link is unaffected. This bit is cleared after the error has been generated.										

15:14	<p><b>GenLinkSel: link select for CRC error generation.</b> Read-Write. Reset: 00b. Selects the link to be used for CRC error injection through GenCrcErrByte1/GenCrcErrByte0.</p> <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>link 0</td> </tr> <tr> <td>01b</td> <td>link 1</td> </tr> <tr> <td>10b</td> <td>link 2</td> </tr> <tr> <td>11b</td> <td>link 3</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Description</u>	00b	link 0	01b	link 1	10b	link 2	11b	link 3														
<u>Bits</u>	<u>Description</u>																								
00b	link 0																								
01b	link 1																								
10b	link 2																								
11b	link 3																								
13:12	<p><b>WDTBaseSel: watchdog timer time base select.</b> Read-write. Reset: 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.</p> <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>1.31 ms</td> </tr> <tr> <td>01b</td> <td>1.28 us</td> </tr> <tr> <td>10b</td> <td>Reserved.</td> </tr> <tr> <td>11b</td> <td>Reserved.</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Description</u>	00b	1.31 ms	01b	1.28 us	10b	Reserved.	11b	Reserved.														
<u>Bits</u>	<u>Description</u>																								
00b	1.31 ms																								
01b	1.28 us																								
10b	Reserved.																								
11b	Reserved.																								
11:9	<p><b>WDTCntSel[2:0]: watchdog timer count select bits[2:0].</b> Read-write. Reset: 0. Selects the count used by the watchdog timer. WDTCntSel = {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:</p> <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>0000b</td> <td>4095</td> </tr> <tr> <td>0001b</td> <td>2047</td> </tr> <tr> <td>0010b</td> <td>1023</td> </tr> <tr> <td>0011b</td> <td>511</td> </tr> <tr> <td>0100b</td> <td>255</td> </tr> <tr> <td>0101b</td> <td>127</td> </tr> <tr> <td>0110b</td> <td>63</td> </tr> <tr> <td>0111b</td> <td>31</td> </tr> <tr> <td>1000b</td> <td>8191</td> </tr> <tr> <td>1001b</td> <td>16383</td> </tr> <tr> <td>1111b-1010b</td> <td>Reserved</td> </tr> </tbody> </table> <p>Because WDTCntSel is split between two registers, care must be taken when programming WDTCntSel to ensure that a reserved value is never used by the watchdog timer or undefined behavior could result.</p>	<u>Bits</u>	<u>Description</u>	0000b	4095	0001b	2047	0010b	1023	0011b	511	0100b	255	0101b	127	0110b	63	0111b	31	1000b	8191	1001b	16383	1111b-1010b	Reserved
<u>Bits</u>	<u>Description</u>																								
0000b	4095																								
0001b	2047																								
0010b	1023																								
0011b	511																								
0100b	255																								
0101b	127																								
0110b	63																								
0111b	31																								
1000b	8191																								
1001b	16383																								
1111b-1010b	Reserved																								
8	<p><b>WDTDis: watchdog timer disable.</b> 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 generated if enabled in D18F3x40 [MCA NB Control].</p>																								
7	<p><b>IoErrDis: IO error response disable.</b> Read-write. Reset: 0. 1=Disables setting either Error bit in link response packets to IO devices on detection of a target or master abort error condition.</p>																								
6	<p><b>CpuErrDis: CPU error response disable.</b> Read-write. Reset: 0. BIOS: 1. 1=Disables generation of a read data error response to the core on detection of a target or master abort error condition.</p>																								
5	<p><b>IoMstAbortDis: IO master abort error response disable.</b> 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[ChgMstAbortToNoErr] are both set, D18F3x180[ChgMstAbortToNoErr] takes precedence.</p>																								

4	<b>SyncPktPropDis: sync packet propagation disable.</b> Read-write. Reset: 0. BIOS: 0. 1=Disables flooding of all outgoing links with sync packets when a sync packet is detected on an incoming link. Sync packets are propagated by default.
3	<b>SyncPktGenDis: sync packet generation disable.</b> Read-write. Reset: 0. BIOS: 0. 1=Disables flooding of all outgoing links with sync packets when a CRC error is detected on an incoming link. By default, sync packet generation for CRC errors is controlled through <a href="#">D18F0x[E4,C4,A4,84]</a> [ <a href="#">Link Control</a> ].
2	<b>SyncFloodOnDramUcEcc: sync flood on uncorrectable DRAM ECC error.</b> Read-write. Reset: 0. BIOS: 1. 1=Enable sync flood of all links with sync packets on detection of an uncorrectable DRAM ECC error.
1	<b>CpuRdDatErrEn: CPU read data error log enable.</b> Read-write. Reset: 0. 1=Enables reporting of read data errors (master aborts and target aborts) for data destined for the CPU on this node. 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

Bits	Description
31:0	<a href="#">MSR0000_0411</a> [31:0] is an alias of <a href="#">D18F3x48</a> . See <a href="#">MSR0000_0411</a> .

#### D18F3x4C MCA NB Status High

Bits	Description
31:0	<a href="#">MSR0000_0411</a> [63:32] is an alias of <a href="#">D18F3x4C</a> . See <a href="#">MSR0000_0411</a> .

#### D18F3x50 MCA NB Address Low

Bits	Description
31:0	<a href="#">MSR0000_0412</a> [31:0] is an alias of <a href="#">D18F3x50</a> . See <a href="#">MSR0000_0412</a> [31:0].

#### D18F3x54 MCA NB Address High

Bits	Description
31:0	<a href="#">MSR0000_0412</a> [63:32] is an alias of <a href="#">D18F3x54</a> . See <a href="#">MSR0000_0412</a> [63:32].

#### D18F3x58 Scrub Rate Control

This register specifies the ECC sequential scrubbing rate for lines of memory and cache. See 2.8.3 [[Memory Scrubbers](#)]. Scrub rates are a platform consideration. See 2.14.1.8 [[Scrub Rate Considerations](#)].



Bits	Description																																																																				
31:29	Reserved.																																																																				
28:24	Reserved.																																																																				
23:5	Reserved.																																																																				
4:0	<p><b>DramScrub: DRAM scrub rate.</b> Read-write. Specifies time between 64 B scrub events. See <a href="#">D18F3x5C</a> and <a href="#">D18F3x60</a>.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00h</td> <td>Disable sequential scrubbing</td> <td>10h</td> <td>1.31 ms</td> </tr> <tr> <td>01h</td> <td>40 ns<sup>1</sup></td> <td>11h</td> <td>2.62 ms</td> </tr> <tr> <td>02h</td> <td>80 ns<sup>1</sup></td> <td>12h</td> <td>5.24 ms</td> </tr> <tr> <td>03h</td> <td>160 ns<sup>1</sup></td> <td>13h</td> <td>10.49 ms</td> </tr> <tr> <td>04h</td> <td>320 ns<sup>1</sup></td> <td>14h</td> <td>20.97 ms</td> </tr> <tr> <td>05h</td> <td>640 ns</td> <td>15h</td> <td>42 ms</td> </tr> <tr> <td>06h</td> <td>1.28 us</td> <td>16h</td> <td>84 ms</td> </tr> <tr> <td>07h</td> <td>2.56 us</td> <td>1Eh-17h</td> <td>Reserved</td> </tr> <tr> <td>08h</td> <td>5.12 us</td> <td>1Fh</td> <td>Reserved</td> </tr> <tr> <td>09h</td> <td>10.2 us</td> <td></td> <td></td> </tr> <tr> <td>0Ah</td> <td>20.5 us</td> <td></td> <td></td> </tr> <tr> <td>0Bh</td> <td>41.0 us</td> <td></td> <td></td> </tr> <tr> <td>0Ch</td> <td>81.9 us</td> <td></td> <td></td> </tr> <tr> <td>0Dh</td> <td>163.8 us</td> <td></td> <td></td> </tr> <tr> <td>0Eh</td> <td>327.7 us</td> <td></td> <td></td> </tr> <tr> <td>0Fh</td> <td>655.4 us</td> <td></td> <td></td> </tr> </tbody> </table> <p>Note:</p> <ol style="list-style-type: none"> <li>This setting is not supported except as a DRAM scrub rate when no other memory accesses are being performed.</li> </ol>	Bits	Description	Bits	Description	00h	Disable sequential scrubbing	10h	1.31 ms	01h	40 ns <sup>1</sup>	11h	2.62 ms	02h	80 ns <sup>1</sup>	12h	5.24 ms	03h	160 ns <sup>1</sup>	13h	10.49 ms	04h	320 ns <sup>1</sup>	14h	20.97 ms	05h	640 ns	15h	42 ms	06h	1.28 us	16h	84 ms	07h	2.56 us	1Eh-17h	Reserved	08h	5.12 us	1Fh	Reserved	09h	10.2 us			0Ah	20.5 us			0Bh	41.0 us			0Ch	81.9 us			0Dh	163.8 us			0Eh	327.7 us			0Fh	655.4 us		
Bits	Description	Bits	Description																																																																		
00h	Disable sequential scrubbing	10h	1.31 ms																																																																		
01h	40 ns <sup>1</sup>	11h	2.62 ms																																																																		
02h	80 ns <sup>1</sup>	12h	5.24 ms																																																																		
03h	160 ns <sup>1</sup>	13h	10.49 ms																																																																		
04h	320 ns <sup>1</sup>	14h	20.97 ms																																																																		
05h	640 ns	15h	42 ms																																																																		
06h	1.28 us	16h	84 ms																																																																		
07h	2.56 us	1Eh-17h	Reserved																																																																		
08h	5.12 us	1Fh	Reserved																																																																		
09h	10.2 us																																																																				
0Ah	20.5 us																																																																				
0Bh	41.0 us																																																																				
0Ch	81.9 us																																																																				
0Dh	163.8 us																																																																				
0Eh	327.7 us																																																																				
0Fh	655.4 us																																																																				

### **D18F3x5C DRAM Scrub Address Low**

In addition to sequential DRAM scrubbing, the DRAM scrubber has a redirect mode for scrubbing DRAM locations accessed during normal operation. This is enabled by setting [D18F3x5C](#)[ScrubReDirEn]. When a DRAM read is generated by any agent other than the DRAM scrubber, correctable ECC errors are corrected as the data is passed to the requestor, but the data in DRAM is not corrected if redirect scrubbing mode is disabled. In scrubber redirect mode, correctable errors detected during normal DRAM read accesses redirect the scrubber to the location of the error. After the scrubber corrects the location in DRAM, it resumes scrubbing from where it left off. DRAM scrub address registers are not modified by the redirect scrubbing mode. Sequential scrubbing and scrubber redirection can be enabled independently or together. ECC errors detected by the scrubber are logged in the MCA registers (See [D18F3x40](#) [MCA NB Control]).

Bits	Description
31:6	<b>ScrubAddr[31:6]: DRAM scrubber address bits[31:6].</b> Read-write; updated-by-hardware. ScrubAddr[47:6] = {D18F3x60[ScrubAddr[47:32]], ScrubAddr[31:6]}. Reset: 0. ScrubAddr points to a DRAM cacheline in physical address space. BIOS should initialize the scrubber address register to the base address of the node specified by D18F1x[17C:140,7C:40] [DRAM Base/Limit] prior to enabling sequential scrubbing through D18F3x58[DramScrub]. When sequential scrubbing is enabled: it starts at the address that the scrubber address registers are initialized to; it increments through address space and updates the scrubber address registers as it does so; when the scrubber reaches the DRAM limit address specified by D18F1x[17C:140,7C:40], it wraps around to the base address. Reads of the scrubber address registers provide the next cacheline to be scrubbed.
5:1	Reserved.
0	<b>ScrubReDirEn: DRAM scrubber redirect enable.</b> Read-write. Reset: 0. BIOS: See Table 47 [DCT Training Specific Register Values]. If a correctable error is discovered from a non-scrubber DRAM read, then the data is corrected before it is returned to the requestor; however, the DRAM location may be left in a corrupted state (until the next time the scrubber address counts up to that location, if sequential scrubbing is enabled through D18F3x58[DramScrub]). 1=Enables the scrubber to immediately scrub any address in which a correctable error is discovered. This bit and sequential scrubbing can be enabled independently or together; if both are enabled, the scrubber jumps from the scrubber address to where the correctable error was discovered, scrubs that location, and then jumps back to where it left off; the scrubber address register is not affected during scrubber redirection.

### D18F3x60 DRAM Scrub Address High

Bits	Description
31:16	Reserved.
15:0	<b>ScrubAddr[47:32]: DRAM scrubber address bits[47:32].</b> See: D18F3x5C[ScrubAddr[31:6]]. Reset: 0.

### D18F3x64 Hardware Thermal Control (HTC)

See 2.10.3.1 [PROCHOT\_L and Hardware Thermal Control (HTC)]. If D18F3xE8[HtcCapable]=0 then this register is reserved.

Bits	Description
31	Reserved.
30:28	<b>HtcPstateLimit: HTC P-state limit select.</b> Read-write. Reset: Product-specific. Specifies the P-state limit of all cores when in the HTC-active state. This field uses hardware P-state numbering and is not changed on a write if the value written is greater than D18F3xDC[HwPstateMaxVal] or less than D18F4x15C[NumBoostStates]. See 2.10.3.1 [PROCHOT_L and Hardware Thermal Control (HTC)] and 2.5.3.1.1.2 [Hardware P-state Numbering].

27:24	<b>HtcHystLmt: HTC hysteresis.</b> Read-write. Reset: Product-specific. The processor exits the HTC-active state when $(Tctl < (HtcTmpLmt - HtcHystLmt))$ . <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>0</td> </tr> <tr> <td>1h</td> <td>0.5</td> </tr> <tr> <td>Eh-2h</td> <td><math>&lt;HtcHystLmt*0.5&gt;</math></td> </tr> <tr> <td>Fh</td> <td>7.5</td> </tr> </tbody> </table>	Bits	Description	0h	0	1h	0.5	Eh-2h	$<HtcHystLmt*0.5>$	Fh	7.5
Bits	Description										
0h	0										
1h	0.5										
Eh-2h	$<HtcHystLmt*0.5>$										
Fh	7.5										
23	<b>HtcSlewSel: HTC slew-controlled temperature select.</b> Read-write. Reset: 0. 1=HTC logic is driven by the slew-controlled temperature, Tctl, specified in <a href="#">D18F3xA4 [Reported Temperature Control]</a> . 0=HTC logic is driven by the measured control temperature with no slew controls.										
22:16	<b>HtcTmpLmt: HTC temperature limit.</b> Read-write. Reset: Product-specific. The processor enters the HTC-active state when Tctl reaches or exceeds the temperature limit defined by this register. <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00h</td> <td>52</td> </tr> <tr> <td>01h</td> <td>52.5</td> </tr> <tr> <td>7Eh-02h</td> <td><math>&lt;(HtcTmpLmt*0.5) + 52&gt;</math></td> </tr> <tr> <td>7Fh</td> <td>115.5</td> </tr> </tbody> </table>	Bits	Description	00h	52	01h	52.5	7Eh-02h	$<(HtcTmpLmt*0.5) + 52>$	7Fh	115.5
Bits	Description										
00h	52										
01h	52.5										
7Eh-02h	$<(HtcTmpLmt*0.5) + 52>$										
7Fh	115.5										
15:8	Reserved.										
7	<b>PslApicLoEn: P-state limit lower value change APIC interrupt enable.</b> Read-write. Reset: 0. PslApicLoEn and PslApicHiEn enable interrupts using <a href="#">APIC330 [LVT Thermal Sensor]</a> of each core when the active P-state limit in <a href="#">MSRC001_0061 [CurPstateLimit]</a> 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	<b>PslApicHiEn: P-state limit higher value change APIC interrupt enable.</b> Read-write. Reset: 0. See PslApicLoEn.										
5	<b>HtcActSts: HTC-active status.</b> 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	<b>HtcAct: HTC-active state.</b> Read-only, updated-by-hardware. 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	<b>HtcEn: HTC enable.</b> Read-write. Reset: 0. BIOS: IF ( <a href="#">D18F3x64 [HtcTmpLmt]</a> ==0) THEN 0 ELSE 1 ENDIF. 1=HTC is enabled; the processor is capable of entering the HTC-active state.										

### D18F3x68 Software P-state Limit

See [2.10.3.3 \[Software P-state Limit Control\]](#). If [D18F3xE8 \[HtcCapable\]](#)=0 then this register is reserved.

Bits	Description
31	Reserved.
30:28	<b>SwPstateLimit: software P-state limit select.</b> Read-write. Reset: Product-specific. Specifies a P-state limit for all cores. Uses hardware P-state numbering; see <a href="#">2.5.3.1.1.2 [Hardware P-state Numbering]</a> . Not changed on a write if the value written is greater than <a href="#">D18F3xDC [HwPstateMaxVal]</a> or less than <a href="#">D18F4x15C [NumBoostStates]</a> . See SwPstateLimitEn.
27:6	Reserved.

5	<b>SwPstateLimitEn: software P-state limit enable.</b> Read-write. Reset: 0. 1=SwPstateLimit is enabled.
4:0	Reserved.

### D18F3x6C Data Buffer Count

Out of cold reset, the processor allocates a minimal number of buffers that is smaller than the default values in the register. BIOS must use **D18F0x6C[RlsLnkFullTokCntImm]** for the values in the register to take effect. This is necessary even if the values are unchanged from the default values.

- To ensure deadlock free operation the following minimum buffer allocations are required:
  - **D18F3x6C[UpRspDBC] >= 1.**
  - **D18F3x6C[DnReqDBC] >= 1.**
  - **D18F3x6C[UpReqDBC] >= 1.**
  - **D18F3x6C[DnRspDBC] >= 1.**
- If **D18F0x[E4,C4,A4,84][IsocEn]=1**: **IsocRspDBC >= 1.**
- The total number of data buffers allocated in this register and **D18F3x7C** must satisfy the following equation:
  - **D18F3x6C[UpReqDBC] + D18F3x6C[UpRspDBC] + D18F3x6C[DnReqDBC] + D18F3x6C[DnRspDBC] + D18F3x6C[IsocRspDBC] + (IF (D18F3x7C[Sri2XbarFreeRspDBC]==0) THEN (D18F3x7C[Sri2XbarFreeXreqDBC]\*2) ELSE D18F3x7C[Sri2XbarFreeXreqDBC] ENDIF) + D18F3x7C[Sri2XbarFreeRspDBC] <= 16.**

Bits	Description
31	Reserved.
30:28	<b>IsocRspDBC: isochronous response data buffer count.</b> Read-write. Reset: 3. BIOS: 1.
27:19	Reserved.
18:16	<b>UpRspDBC: upstream response data buffer count.</b> Read-write. Reset: 2. BIOS: 1.
15	Reserved.
14:8	Reserved.
7:6	<b>DnRspDBC: downstream response data buffer count.</b> Read-write. Reset: 2. BIOS: 1.
5:4	<b>DnReqDBC: downstream request data buffer count.</b> Read-write. Reset: 1. BIOS: 1.
3	Reserved.
2:0	<b>UpReqDBC: upstream request data buffer count.</b> Read-write. Reset: 2. BIOS: 2.

### D18F3x70 SRI to XBAR Command Buffer Count

Reset: 1111\_2153h. Out of cold reset, the processor allocates a minimal number of buffers that is smaller than the default values in the register. BIOS must use **D18F0x6C[RlsLnkFullTokCntImm]** for the values in the register to take effect. This is necessary even if the values are unchanged from the default values.

- To ensure deadlock free operation the following minimum buffer allocations are required:
  - **D18F3x70[UpRspCBC] >= 1.**
  - **D18F3x70[UpPreqCBC] >= 1.**
  - **D18F3x70[DnPreqCBC] >= 1.**
  - **D18F3x70[UpReqCBC] >= 1.**
  - **D18F3x70[DnReqCBC] >= 1.**
  - **D18F3x70[DnRspCBC] >= 1.**
- If any of the **D18F0x[E4,C4,A4,84][IsocEn]** bits are set:

IsocReqCBC >= 1                      IsocRspCBC >= 1

- If D18F0x[E4,C4,A4,84][IsocEn]=1 and isochronous posted requests may be generated by the system:  
IsocPreqCBC >= 1
- The total number of SRI to XBAR command buffers allocated in this register and D18F3x7C must satisfy the following equation:
  - $D18F3x70[IsocRspCBC] + D18F3x70[IsocPreqCBC] + D18F3x70[IsocReqCBC] + D18F3x70[UpRspCBC] + D18F3x70[DnPreqCBC] + D18F3x70[UpPreqCBC] + D18F3x70[DnReqCBC] + D18F3x70[DnRspCBC] + D18F3x70[UpReqCBC] + D18F3x7C[Sri2XbarFreeRspCBC] + D18F3x7C[Sri2XbarFreeXreqCBC] \leq 24$ .

Bits	Description
31	Reserved.
30:28	<b>IsocRspCBC: isoc response command buffer count.</b> Read-write. BIOS: 1.
27	Reserved.
26:24	<b>IsocPreqCBC: isoc posted request command buffer count.</b> Read-write. BIOS: 0.
23	Reserved.
22:20	<b>IsocReqCBC: isoc request command buffer count.</b> Read-write. BIOS: 1.
19	Reserved.
18:16	<b>UpRspCBC: upstream response command buffer count.</b> Read-write. BIOS: 3.
15	Reserved.
14:12	<b>DnPreqCBC: downstream posted request command buffer count.</b> Read-write. BIOS: 1.
11	Reserved.
10:8	<b>UpPreqCBC: upstream posted request command buffer count.</b> Read-write. BIOS: 1.
7:6	<b>DnRspCBC: downstream response command buffer count.</b> Read-write. BIOS: 1.
5:4	<b>DnReqCBC: downstream request command buffer count.</b> Read-write. BIOS: 1.
3	Reserved.
2:0	<b>UpReqCBC: upstream request command buffer count.</b> Read-write. BIOS: 3

**D18F3x74 XBAR to SRI Command Buffer Count**

Reset: 0007\_1111h. Out of cold reset, the processor allocates a minimal number of buffers that is smaller than the default values in the register. BIOS must use D18F0x6C[RlsLnkFullTokCntImm] for the values in the register to take effect. This is necessary even if the values are unchanged from the default values.

**Table 173: Buffer Definitions**

Term	Definition
<b>SpqSize</b>	Probe command queue size. SpqSize = 20.
<b>SrqSize</b>	SRQ (XBAR command and probe response to SRI) queue size. SrqSize = 36.

**Table 173: Buffer Definitions**

Term	Definition
<b>PrbRsp</b>	SRQ entries hard allocated to probe responses. PrbRsp = 4.
<b>MpbcSize</b>	MPB command buffer size. MpbcSize = 24.
<b>McqSize</b>	MCT command queue size. McqSize = 36.

- To ensure deadlock free operation the following minimum buffer allocations are required:
  - $D18F3x74[ProbeCBC] \geq 2$ .
  - $D18F3x74[UpReqCBC] \geq 1$ .
  - $D18F3x74[UpPreqCBC] \geq 1$ .
  - $(IsocReqCBC + IsocPreqCBC + DRReqCBC) \leq 31$ .
  - $(IsocReqCBC + IsocPreqCBC + DRReqCBC) \leq (McqSize - 16)$ .
- If any of  $D18F0x[E4,C4,A4,84][IsocEn]$  bits are set, then  $IsocReqCBC \geq 1$ .
- If any of the  $D18F0x[E4,C4,A4,84][IsocEn]$  bits are set and isochronous posted requests may be generated by the system:  
 $IsocPreqCBC \geq 1$
- The total number of XBAR to SRI command buffers allocated in this register and  $D18F3x7C$  must satisfy the following equation:
  - $D18F3x74[UpReqCBC] + D18F3x74[UpPreqCBC] + D18F3x74[DnReqCBC] + D18F3x74[DnPreqCBC] + D18F3x74[IsocReqCBC] + D18F3x74[IsocPreqCBC] + D18F3x74[DRReqCBC] + D18F3x7C[Xbar2SriFreeListCBC] + (D18F3x1A0[CpuCmdBufCnt] * NumOfCompUnitsOnNode) + D18F3x1A0[CpuToNbFreeBufCnt] + PrbRsp \leq SrqSize$
- The total number of SPQ (probe command) buffers allocated must satisfy the following equation:
  - $(D18F3x17C[SPQPrbFreeCBC] + D18F3x74[ProbeCBC]) \leq SpqSize$ .

Bits	Description								
31:28	<b>DRReqCBC: display refresh request command buffer count.</b> Read-write. BIOS: 0.								
27	Reserved.								
26:24	<b>IsocPreqCBC: isochronous posted request command buffer count.</b> Read-write. BIOS: 1.								
23:20	<b>IsocReqCBC: isochronous request command buffer count.</b> Read-write. BIOS: 1.								
19:16	<b>ProbeCBC: probe command buffer count.</b> Read-write. BIOS: Ch. <table border="0" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; border-bottom: 1px solid black;">Bits</th> <th style="text-align: left; border-bottom: 1px solid black;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>0 buffers</td> </tr> <tr> <td>Ch-1h</td> <td>&lt;ProbeCBC&gt; buffers</td> </tr> <tr> <td>Fh-Dh</td> <td>Reserved.</td> </tr> </tbody> </table>	Bits	Description	0h	0 buffers	Ch-1h	<ProbeCBC> buffers	Fh-Dh	Reserved.
Bits	Description								
0h	0 buffers								
Ch-1h	<ProbeCBC> buffers								
Fh-Dh	Reserved.								
15	Reserved.								

14:12	<b>DnPreqCBC: downstream posted request command buffer count.</b> Read-write. BIOS: 0.
11	Reserved.
10:8	<b>UpPreqCBC: upstream posted request command buffer count.</b> Read-write. BIOS: 1.
7	Reserved.
6:4	<b>DnReqCBC: downstream request command buffer count.</b> Read-write. BIOS: 0.
3	Reserved.
2:0	<b>UpReqCBC: upstream request command buffer count.</b> Read-write. BIOS: 1.

**D18F3x78 MCT to XBAR Buffer Count**

Reset: 0016\_0519h. Out of cold reset, the processor allocates a minimal number of buffers that is smaller than the default values in the register. BIOS must use **D18F0x6C**[RIsLnkFullTokCntImm] for the values in the register to take effect. This is necessary even if the values are unchanged from the default values.

- To ensure deadlock free operation the following minimum buffer allocations are required:  
 $ProbeCBC \geq 1$                        $RspCBC \geq 1$                        $RspDBC \geq 2$   
 $RspDBC \geq D18F2x11C[MctPrefReqLimit]+2$
- The total number of command buffers allocated in this register must satisfy the following equation:  
 $(D18F3x78[ProbeCBC] + D18F3x78[RspCBC]) \leq MpbcsSize$ .

Bits	Description												
31:22	Reserved.												
21:16	<b>RspDBC: response data buffer count.</b> Read-write. BIOS: 20h. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>01h-00h</td> <td>Reserved</td> </tr> <tr> <td>02h</td> <td>2 Buffers</td> </tr> <tr> <td>1Fh-03h</td> <td>&lt;RspDBC&gt; Buffers</td> </tr> <tr> <td>20h</td> <td>32 Buffers</td> </tr> <tr> <td>3Fh-21h</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Description	01h-00h	Reserved	02h	2 Buffers	1Fh-03h	<RspDBC> Buffers	20h	32 Buffers	3Fh-21h	Reserved
Bits	Description												
01h-00h	Reserved												
02h	2 Buffers												
1Fh-03h	<RspDBC> Buffers												
20h	32 Buffers												
3Fh-21h	Reserved												
15:13	Reserved.												
12:8	<b>ProbeCBC: probe command buffer count.</b> Read-write. BIOS: 8h.												
7:6	Reserved.												
5:0	<b>RspCBC: response command buffer count.</b> Read-write. BIOS: 10h.												

### D18F3x7C Free List Buffer Count

Reset: 0003\_660Ch. Out of cold reset, the processor allocates a minimal number of buffers that is smaller than the default values in the register. BIOS must use **D18F0x6C[RlsLnkFullTokCntImm]** for the values in the register to take effect. This is necessary even if the values are unchanged from the default values. See **D18F3x6C** and **D18F3x70**.

- To ensure deadlock free operation the following minimum buffer allocations are required:
  - IF (**D18F3x7C[Sri2XbarFreeRspCBC]**==0) THEN (**D18F3x7C[Sri2XbarFreeXreqCBC]**>2).
  - IF (**D18F3x7C[Sri2XbarFreeRspCBC]**!=0) THEN (**D18F3x7C[Sri2XbarFreeRspCBC]**>2).
  - IF (**D18F3x7C[Sri2XbarFreeRspDBC]**==0) THEN (**D18F3x7C[Sri2XbarFreeXreqDBC]**>2).
  - IF (**D18F3x7C[Sri2XbarFreeRspDBC]**!=0) THEN (**D18F3x7C[Sri2XbarFreeRspDBC]**>2).
  - **D18F3x7C[Xbar2SriFreeListCBC]** >= (**D18F3x1A0[CpuToNbFreeBufCnt]** \* **NumOfCompUnitsOnNode**) + 2.

Bits	Description
31	Reserved.
30:28	<b>Xbar2SriFreeListCBInc: XBAR to SRI free list command buffer increment.</b> Read-write. This field is use to add buffers to the free list pool if they are reclaimed from hard allocated entries without having to go through warm reset. This field may only be programmed after buffers have been allocated and released via <b>D18F0x6C[RlsLnkFullTokCntImm]</b> .
27:23	Reserved.
22:20	<b>Sri2XbarFreeRspDBC: SRI to XBAR free response data buffer count.</b> Read-write. BIOS: 0.
19:16	<b>Sri2XbarFreeXreqDBC: SRI to XBAR free request and posted request data buffer count.</b> Read-write. BIOS: 5h. If <b>Sri2XbarFreeRspDBC</b> =0h, then these buffers are shared between requests, responses and posted requests and the number of buffers allocated is two times the value of this field.
15:12	<b>Sri2XbarFreeRspCBC: SRI to XBAR free response command buffer count.</b> Read-write. BIOS: 0h.
11:8	<b>Sri2XbarFreeXreqCBC: SRI to XBAR free request and posted request command buffer count.</b> Read-write. BIOS: 6h. If <b>Sri2XbarFreeRspCBC</b> =0h, then these buffers are shared between requests, responses and posted requests and the number of buffers allocated is two times the value of this field.
7:6	Reserved.
5:0	<b>Xbar2SriFreeListCBC: XBAR to SRI free list command buffer count.</b> Read-write. BIOS: 1Bh.

### D18F3x[84:80] ACPI Power State Control

This block consists 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 descriptions below for the associated ACPI state and system management actions for each of the 8 SMAF codes. The **SmafAct** fields specify the system management actions taken when the corresponding SMAF code is received. For instance, a SMAF code of 5 results in the power management actions specified by **SmafAct5**. Some ACPI states and associated SMAF codes may not be supported in certain conditions. See 2.5 [Power Management] for which states are supported.



When a link STPCLK assertion command is received by the processor, the power management commands specified by the register with the corresponding SMAF code are invoked. When the STPCLK deassertion command is received by the processor, the processor returns into the operational state.

In multi-node systems, these registers should be programmed identically in all nodes.

**Table 174: SMAF Action Definition**

Register	SmafAct	ACPI state	Description
D18F3x84[31:24]	SmafAct7	C1	Initiated when a Halt instruction is executed by processor. This does not involve the interaction with the SMC, therefore the SMC is required to never send STPCLK assertion commands with SMAF=7h.
D18F3x84[23:16]	SmafAct6	S4/S5	Initiated by a processor access to the ACPI-defined PM1_CNTa register.
D18F3x84[15:8]	SmafAct5	Throttling	Occurs based upon SMC hardware-initiated throttling. AMD recommends using PROCHOT_L for thermal throttling and not implementing stop clock based throttling.
D18F3x84[7:0]	SmafAct4	S3	Initiated by a processor access to the ACPI-defined PM1_CNTa register.
D18F3x80[31:24]	SmafAct3	S1	Initiated by a processor access to the ACPI-defined PM1_CNTa register.
D18F3x80[23:16]	SmafAct2	-	
D18F3x80[15:8]	SmafAct1	C1E, or Link init.	Initiated by an access to the ACPI-defined P_LVL3 register.
D18F3x80[7:0]	SmafAct0	C2	Initiated by a processor access to the ACPI-defined P_LVL2 register.

**D18F3x80 ACPI Power State Control Low**

Reset: 0000\_0000h. Read-write.

Bits	Description
31:29	<b>ClkDivisorSmafAct3.</b> See: ClkDivisorSmafAct0.
28:27	Reserved.
26	<b>NbGateEnSmafAct3.</b> See: NbGateEnSmafAct0.
25	<b>NbLowPwrEnSmafAct3.</b> See: NbLowPwrEnSmafAct0.
24	<b>CpuPrbEnSmafAct3.</b> See: CpuPrbEnSmafAct0.
23:21	<b>ClkDivisorSmafAct2.</b> See: ClkDivisorSmafAct0.
20:19	Reserved.
18	<b>NbGateEnSmafAct2.</b> See: NbGateEnSmafAct0.
17	<b>NbLowPwrEnSmafAct2.</b> See: NbLowPwrEnSmafAct0.
16	<b>CpuPrbEnSmafAct2.</b> See: CpuPrbEnSmafAct0.
15:13	<b>ClkDivisorSmafAct1.</b> See: ClkDivisorSmafAct0.

12:11	Reserved.																				
10	<b>NbGateEnSmafAct1.</b> See: NbGateEnSmafAct0.																				
9	<b>NbLowPwrEnSmafAct1.</b> See: NbLowPwrEnSmafAct0.																				
8	<b>CpuPrbEnSmafAct1.</b> See: CpuPrbEnSmafAct0.																				
7:5	<p><b>ClkDivisorSmafAct0: clock divisor.</b> Read-write.                  Specifies the core clock frequency while in the low-power state. This divisor is relative to the current FID frequency, or:</p> <ul style="list-style-type: none"> <li>100 MHz * (10h + MSRC001_00[6B:64][CpuFid]) of the current P-state specified by MSRC001_0063[CurPstate].</li> </ul> <p>If MSRC001_00[6B:64][CpuDid] of the current P-state indicates a divisor that is deeper than specified by this field, then no frequency change is made when entering the low-power state associated with this register.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>/1</td> <td>100b</td> <td>/16</td> </tr> <tr> <td>001b</td> <td>/2</td> <td>101b</td> <td>/128</td> </tr> <tr> <td>010b</td> <td>/4</td> <td>110b</td> <td>/512</td> </tr> <tr> <td>011b</td> <td>/8</td> <td>111b</td> <td>Turn off clocks</td> </tr> </tbody> </table>	Bits	Description	Bits	Description	000b	/1	100b	/16	001b	/2	101b	/128	010b	/4	110b	/512	011b	/8	111b	Turn off clocks
Bits	Description	Bits	Description																		
000b	/1	100b	/16																		
001b	/2	101b	/128																		
010b	/4	110b	/512																		
011b	/8	111b	Turn off clocks																		
4:3	Reserved.																				
2	<b>NbGateEnSmafAct0: northbridge gate enable.</b> Read-write. This bit does not control hardware. NbLowPwrEn is required to be set if this bit is set.																				
1	<b>NbLowPwrEnSmafAct0: Northbridge low-power enable.</b> Read-write. 1=The NB clock is ramped down to the divisor specified by D18F3xD4[NbClkDiv] and DRAM is placed into self-refresh mode when LDTSTOP_L is asserted while in the low-power state.																				
0	<p><b>CpuPrbEnSmafAct0: CPU direct probe enable.</b> Read-write. Specifies how probes are handled while in the low-power state. 0=When the probe request comes into the NB, the core clock is brought up to the COF (based on the current P-state), all outstanding probes are completed, the core waits for a hysteresis time based on D18F3xD4[ClkRampHystSel], and then the core clock is brought down to the frequency specified by ClkDivisor. 1=The core clock does not change frequency; the probe is handled at the frequency specified by ClkDivisor; this may only be set if:</p> <ul style="list-style-type: none"> <li>ClkDivisor specifies a divide-by 1, 2, 4, 8, or 16 and NbCof &lt;= 3.2 GHz</li> <li>ClkDivisor specifies a divide-by 1, 2, 4, or 8 and NbCof &gt;= 3.4 GHz</li> </ul> <p>This bit also specifies functionality of the timer used for cache flushing during halt. See D18F3xDC[CacheFlushOnHaltTmr].</p> <ul style="list-style-type: none"> <li>If ((D18F3x84[CpuPrbEnSmafAct7]==0) &amp;&amp; (D18F3xDC[IgnCpuPrbEn]==0)), only the time when the core is halted and has its clocks ramped up to service probes is counted.</li> <li>If ((D18F3x84[CpuPrbEnSmafAct7]==1) or (D18F3xDC[IgnCpuPrbEn]==1)), all of the time the core is halted is counted.</li> </ul>																				

**D18F3x84 ACPI Power State Control High**

Reset: 0000\_0000h. Read-write.

Bits	Description
31:29	<b>ClkDivisorSmafAct7.</b> See: D18F3x80[ClkDivisorSmafAct0].
28:27	Reserved.
26	<b>NbGateEnSmafAct7.</b> See: D18F3x80[NbGateEnSmafAct0].
25	<b>NbLowPwrEnSmafAct7.</b> See: D18F3x80[NbLowPwrEnSmafAct0].

24	<b>CpuPrbEnSmafAct7</b> . See: <a href="#">D18F3x80[CpuPrbEnSmafAct0]</a> .
23:21	<b>ClkDivisorSmafAct6</b> . See: <a href="#">D18F3x80[ClkDivisorSmafAct0]</a> .
20:19	Reserved.
18	<b>NbGateEnSmafAct6</b> . See: <a href="#">D18F3x80[NbGateEnSmafAct0]</a> .
17	<b>NbLowPwrEnSmafAct6</b> . See: <a href="#">D18F3x80[NbLowPwrEnSmafAct0]</a> .
16	<b>CpuPrbEnSmafAct6</b> . See: <a href="#">D18F3x80[CpuPrbEnSmafAct0]</a> .
15:13	<b>ClkDivisorSmafAct5</b> . See: <a href="#">D18F3x80[ClkDivisorSmafAct0]</a> .
12:11	Reserved.
10	<b>NbGateEnSmafAct5</b> . See: <a href="#">D18F3x80[NbGateEnSmafAct0]</a> .
9	<b>NbLowPwrEnSmafAct5</b> . See: <a href="#">D18F3x80[NbLowPwrEnSmafAct0]</a> .
8	<b>CpuPrbEnSmafAct5</b> . See: <a href="#">D18F3x80[CpuPrbEnSmafAct0]</a> .
7:5	<b>ClkDivisorSmafAct4</b> . See: <a href="#">D18F3x80[ClkDivisorSmafAct0]</a> .
4:3	Reserved.
2	<b>NbGateEnSmafAct4</b> . See: <a href="#">D18F3x80[NbGateEnSmafAct0]</a> .
1	<b>NbLowPwrEnSmafAct4</b> . See: <a href="#">D18F3x80[NbLowPwrEnSmafAct0]</a> . BIOS: 1.
0	<b>CpuPrbEnSmafAct4</b> . See: <a href="#">D18F3x80[CpuPrbEnSmafAct0]</a> .

#### D18F3x88 NB Configuration 1 Low (NB\_CFG1\_LO)

Bits	Description
31:0	<a href="#">MSRC001_001F</a> [31:0] is an alias of <a href="#">D18F3x88</a> . See <a href="#">MSRC001_001F</a> .

#### D18F3x8C NB Configuration 1 High (NB\_CFG1\_HI)

Bits	Description
31:0	<a href="#">MSRC001_001F</a> [63:32] is an alias of <a href="#">D18F3x8C</a> . See <a href="#">MSRC001_001F</a> .

#### D18F3xA0 Power Control Miscellaneous

Bits	Description
31	<b>CofVidProg: COF and VID of P-states programmed.</b> Read-only. Reset: Product-specific. 1=Out of cold reset, the VID, FID, and DID values of the P-state registers specified by <a href="#">MSRC001_0071</a> [StartupPstate] and <a href="#">D18F5x174</a> [StartupNbPstate] have been applied to the processor. 0=Out of cold reset, the boot VID is applied to all processor power planes, the NB clock plane is set to 800 MHz (with a FID of 00h=800 MHz and a DID of 0b) and core CPU clock planes are set to 800 MHz (with a FID of 00h=1.6 GHz and a DID of 1h). Registers containing P-state information such as FID, DID, and VID values are valid out of cold reset independent of the state of <a href="#">D18F3xA0</a> [CofVidProg]. BIOS must transition the processor to a valid P-state out of cold reset when <a href="#">D18F3xA0</a> [CofVidProg]=0. See <a href="#">2.5.3.1.6 [BIOS Requirements for Core P-state Initialization and Transitions]</a> .
30:28	Reserved.

27:16	<b>ConfigId: Configuration identifier.</b> Read-only. Reset: Product-specific. Specifies the configuration ID associated with the product.																				
15	Reserved.																				
14	<b>Svi2HighFreqSel: SVI2 high frequency select.</b> Read-write. Cold reset: 0. BIOS: 1. 0=3.4 MHz. 1=20 MHz. Writes to this field take effect at the next SVI command boundary. If 20 MHz is supported by the VRM, BIOS should program this to 1 prior to any VID transitions. Once this bit is set, it should not be cleared until the next cold reset.																				
13:11	<p><b>PllLockTime: PLL synchronization lock time.</b> Read-write. Reset: 0. BIOS: 001b. If a P-state change occurs that applies a new FID to the PLL, this field specifies the time required for the PLL to lock to the new frequency.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>1 us</td> <td>100b</td> <td>8 us</td> </tr> <tr> <td>001b</td> <td>2 us</td> <td>101b</td> <td>16 us</td> </tr> <tr> <td>010b</td> <td>3 us</td> <td>110b</td> <td>Reserved</td> </tr> <tr> <td>011b</td> <td>4 us</td> <td>111b</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Description	Bits	Description	000b	1 us	100b	8 us	001b	2 us	101b	16 us	010b	3 us	110b	Reserved	011b	4 us	111b	Reserved
Bits	Description	Bits	Description																		
000b	1 us	100b	8 us																		
001b	2 us	101b	16 us																		
010b	3 us	110b	Reserved																		
011b	4 us	111b	Reserved																		
10	Reserved.																				
9	Reserved.																				
8	<b>PsiVid[7].</b> Read-write. Reset: 0. BIOS: 2.5.1.3.1.1. See PsiVid[6:0].																				
7	<b>PsiVidEn: PSI_L VID enable.</b> Read-write. Reset: 0. BIOS: 2.5.1.3.1.1. This bit specifies how PSI_L is controlled. This signal may be used by the voltage regulator to improve efficiency while in reduced power states. 1=Control over the PSI_L signal is as specified by the PsiVid field of this register. 0=PSI_L is always high. See 2.5.1.3.1 [PSIx_L Bit].																				
6:0	<b>PsiVid[6:0]: PSI_L VID threshold.</b> Read-write. Reset: 0. BIOS: 2.5.1.3.1.1. PsiVid[7:0] = {PsiVid[7], PsiVid[6:0]}. When enabled by PsiVidEn, PsiVid[7:0] specifies the threshold value of the VID code generated by the processor, which in turn determines the state of PSI0_L. When the VID code generated by the processor is less than PsiVid[7:0] (i.e., the VID code is specifying a higher voltage level than the PsiVid-specified voltage level), then PSI0_L is high; when the VID code is greater than or equal to PsiVid[7:0], PSI0_L is driven low. See 2.5.1.3.1 [PSIx_L Bit].																				

### D18F3x4 Reported Temperature Control

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	<p><b>CurTmp: current temperature.</b>  IF (D18F3xA4[CurTmpTjSel]==11b) THEN Read-write. ELSE Read-only, updated-by-hardware. ENDIF. Reset: X. Provides the current control temperature, Tctl, after the slew-rate controls have been applied.  RangeUnadjusted = (D18F3xA4[CurTmpTjSel]!=11b).</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>RangeUnadjusted</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>000h</td> <td>0</td> <td>-49</td> </tr> <tr> <td>001h</td> <td>0</td> <td>-48.875</td> </tr> <tr> <td>7FEh-002h</td> <td>0</td> <td>&lt;(CurTmp*0.125)-49&gt;</td> </tr> <tr> <td>7FFh</td> <td>0</td> <td>206.875</td> </tr> <tr> <td>000h</td> <td>1</td> <td>0</td> </tr> <tr> <td>001h</td> <td>1</td> <td>0.125</td> </tr> <tr> <td>7FEh-002h</td> <td>1</td> <td>&lt;CurTmp*0.125&gt;</td> </tr> <tr> <td>7FFh</td> <td>1</td> <td>255.875</td> </tr> </tbody> </table>	Bits	RangeUnadjusted	Description	000h	0	-49	001h	0	-48.875	7FEh-002h	0	<(CurTmp*0.125)-49>	7FFh	0	206.875	000h	1	0	001h	1	0.125	7FEh-002h	1	<CurTmp*0.125>	7FFh	1	255.875
Bits	RangeUnadjusted	Description																										
000h	0	-49																										
001h	0	-48.875																										
7FEh-002h	0	<(CurTmp*0.125)-49>																										
7FFh	0	206.875																										
000h	1	0																										
001h	1	0.125																										
7FEh-002h	1	<CurTmp*0.125>																										
7FFh	1	255.875																										
20	Reserved.																											
19:18	Reserved.																											
17:16	<p><b>CurTmpTjSel: Current temperature select.</b> Read-write. Reset: 00. These bits may be used for diagnostic software.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>CurTmp provides the read-only Tctl value.</td> </tr> <tr> <td>01b</td> <td>Reserved.</td> </tr> <tr> <td>10b</td> <td>Reserved.</td> </tr> <tr> <td>11b</td> <td>CurTmp is a read-write register that specifies a value used to create Tctl. The two LSB's are read-only zero.</td> </tr> </tbody> </table>	Bits	Description	00b	CurTmp provides the read-only Tctl value.	01b	Reserved.	10b	Reserved.	11b	CurTmp is a read-write register that specifies a value used to create Tctl. The two LSB's are read-only zero.																	
Bits	Description																											
00b	CurTmp provides the read-only Tctl value.																											
01b	Reserved.																											
10b	Reserved.																											
11b	CurTmp is a read-write register that specifies a value used to create Tctl. The two LSB's are read-only zero.																											
15:13	Reserved.																											
12:8	<p><b>PerStepTimeDn: per step time down.</b> Read-write. Cold reset: 18h. BIOS: 0Fh. Specifies the time that measured temperature must remain below Tctl before applying a 0.125 downward step. See: PerStepTimeUp.</p>																											
7	<p><b>TmpSlewDnEn: temperature slew downward enable.</b> Read-write. Cold reset: 0. BIOS: 1. 1=Downward slewing enabled. 0=Downward slewing disabled.</p>																											
6:5	<p><b>TmpMaxDiffUp: temperature maximum difference up.</b> Read-write. Cold reset: 00b. BIOS: 11b. Specifies the maximum difference, (measured temperature - Tctl), when Tctl immediately updates to the measured temperature.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>0.0 (disable upward slew)</td> </tr> <tr> <td>01b</td> <td>1.0</td> </tr> <tr> <td>10b</td> <td>3.0</td> </tr> <tr> <td>11b</td> <td>9.0</td> </tr> </tbody> </table>	Bits	Description	00b	0.0 (disable upward slew)	01b	1.0	10b	3.0	11b	9.0																	
Bits	Description																											
00b	0.0 (disable upward slew)																											
01b	1.0																											
10b	3.0																											
11b	9.0																											
4:0	<p><b>PerStepTimeUp: per 1/8th degree step time up.</b> Read-write. Cold reset: 00h. BIOS: 0Fh. Specifies the time that measured temperature must remain above Tctl before applying a 0.125 upward step.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>1Fh-00h</td> <td>&lt;(PerStepTimeUp[2:0] + 1) * 10^PerStepTimeUp[4:3]&gt; ms, ranging from 1 ms to 8000 ms.</td> </tr> </tbody> </table>	Bits	Definition	1Fh-00h	<(PerStepTimeUp[2:0] + 1) * 10^PerStepTimeUp[4:3]> ms, ranging from 1 ms to 8000 ms.																							
Bits	Definition																											
1Fh-00h	<(PerStepTimeUp[2:0] + 1) * 10^PerStepTimeUp[4:3]> ms, ranging from 1 ms to 8000 ms.																											

**D18F3xA8 Pop Up and Down P-states**

Bits	Description
31:29	<b>PopDownPstate.</b> Read-write. Reset: <a href="#">D18F3xDC[HwPstateMaxVal]</a> . BIOS: <a href="#">D18F3xDC[HwPstateMaxVal]</a> . Specifies the pop-down P-state number. This field uses hardware P-state numbering. See <a href="#">2.5.3.2.3.3 [Core C6 (CC6) State]</a> .
28:0	Reserved.

**D18F3xB8 NB Array Address**

Reset: xxxx\_xxxxh. [D18F3xB8 \[NB Array Address\]](#) and [D18F3xBC \[NB Array Data Port\]](#) provide a mechanism to inject errors into DRAM and data read from internal NB arrays.

[D18F3xB8](#) should first be written with the target array and address within the array. Read and write accesses to [D18F3xBC](#) then access the target address within the target array.

Bits	Description								
31:28	<b>ArraySelect.</b> Read-write. Selects the NB array to access. <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>7h-0h</td> <td>Reserved</td> </tr> <tr> <td>8h</td> <td><a href="#">D18F3xBC_x8 [DRAM ECC]</a></td> </tr> <tr> <td>Fh-9h</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Description	7h-0h	Reserved	8h	<a href="#">D18F3xBC_x8 [DRAM ECC]</a>	Fh-9h	Reserved
Bits	Description								
7h-0h	Reserved								
8h	<a href="#">D18F3xBC_x8 [DRAM ECC]</a>								
Fh-9h	Reserved								
27:10	Reserved.								
9:0	<b>ArrayAddress.</b> Read-write. Selects the location to access within the selected array. This format of this field is a function of ArraySelect. <table border="1"> <thead> <tr> <th>ArraySelect</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>8h</td> <td>DRAM ECC.</td> </tr> <tr> <td>All others</td> <td>Reserved</td> </tr> </tbody> </table>	ArraySelect	Description	8h	DRAM ECC.	All others	Reserved		
ArraySelect	Description								
8h	DRAM ECC.								
All others	Reserved								

**D18F3xBC NB Array Data Port**

See [D18F3xB8](#) for register access information. Address: [D18F3xB8\[ArraySelect\]](#).

Bits	Description
31:0	<b>Data.</b>

**D18F3xBC\_x8 DRAM ECC**

This register controls injection of errors in writes to DRAM. See [2.14.3.1 \[DRAM Error Injection\]](#).

Bits	Description
31:29	Reserved.

28:20	<b>ErrInjEn: enable error injection to word.</b> Read-write. Reset: 0. Each bit in this field corresponds to a 16-bit DRAM word and enables injecting errors in that word. <table border="1"> <thead> <tr> <th>Bit</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>[0]</td> <td>Data[15:0]</td> </tr> <tr> <td>[1]</td> <td>Data[31:16]</td> </tr> <tr> <td>[2]</td> <td>Data[47:32]</td> </tr> <tr> <td>[3]</td> <td>Data[63:48]</td> </tr> <tr> <td>[4]</td> <td>Data[79:64]</td> </tr> <tr> <td>[5]</td> <td>Data[95:80]</td> </tr> <tr> <td>[6]</td> <td>Data[111:96]</td> </tr> <tr> <td>[7]</td> <td>Data[127:112]</td> </tr> <tr> <td>[8]</td> <td>ECC[15:0]</td> </tr> </tbody> </table>	Bit	Description	[0]	Data[15:0]	[1]	Data[31:16]	[2]	Data[47:32]	[3]	Data[63:48]	[4]	Data[79:64]	[5]	Data[95:80]	[6]	Data[111:96]	[7]	Data[127:112]	[8]	ECC[15:0]
Bit	Description																				
[0]	Data[15:0]																				
[1]	Data[31:16]																				
[2]	Data[47:32]																				
[3]	Data[63:48]																				
[4]	Data[79:64]																				
[5]	Data[95:80]																				
[6]	Data[111:96]																				
[7]	Data[127:112]																				
[8]	ECC[15:0]																				
19	Reserved.																				
18	<b>DramErrEn.</b> Read-write. Reset: 0. 1=Errors are continually injected on DRAM writes. The error injection takes place only on DRAM write accesses and should be initiated by a non-cacheable store. Errors continue to be injected on writes until this bit is cleared to a 0 by software.																				
17	<b>EccWrReq.</b> Read; write-1-only; cleared-by-hardware. Reset: 0. 1=Error is injected on DRAM write at the bits enabled by ErrInjEn and EccVector. A single error injection takes place on the next DRAM write access and should be initiated by a non-cacheable store. This bit is cleared by hardware after the write.																				
16	<b>EccRdReq.</b> Read; write-1-only; cleared-by-hardware. Reset: 0. 1=Indicates a DRAM ECC read is requested. The read takes place on the next DRAM read access and should be initiated by a non-cacheable load. The ECC bits read from DRAM are stored in EccVector. This bit is cleared by hardware after the read.																				
15:0	<b>EccVector: error injection vector.</b> Read-write. Reset: x. When used in conjunction with EccWrReq, each bit of EccVector enables injecting errors to the corresponding bit within each word enabled by ErrInjEn. When used in conjunction with EccRdReq, EccVector holds the contents of the DRAM ECC bits after the read.																				

### D18F3xD4 Clock Power/Timing Control 0

Bits	Description
31	<b>NbClkDivApplyAll.</b> Read-write. Cold reset: 0. BIOS: 1. See NbClkDiv.

30:28	<p><b>NbClkDiv: NB clock divisor.</b> Read-write. Cold reset: Product-specific. BIOS: 001b.</p> <p>Specifies the NB CLK divisor associated with <a href="#">D18F3x80/D18F3x84</a>[NbLowPwrEn]. This divisor is applied while LDTSTOP_L is asserted if the corresponding core CLK divisor, <a href="#">D18F3x80/D18F3x84</a>[ClkDivisor], is set to “turn off clocks” or if NBClkDivApplyAll=1; otherwise, the divisor specified by <a href="#">D18F3x80/D18F3x84</a>[ClkDivisor] is applied. This divisor is relative to the current NB FID frequency, or:</p> <p style="padding-left: 20px;">100 MHz * (4 + <a href="#">D18F5x16</a>[C:0][NbFid]).</p> <p>If <a href="#">D18F5x16</a>[C:0][NbDid] of the current NB P-state indicates a divisor that is lower than specified by this field, then no NB frequency change is made when entering the low-power state associated with this register (i.e., if this field specifies a divide-by 1 and the DID is divide-by 2, then the divisor remains 2 while in the low-power state). This field is encoded as follows:</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">Bits</th> <th style="text-align: left;">Description</th> <th style="text-align: left;">Bits</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>Divide-by 1</td> <td>100b</td> <td>Divide-by 16</td> </tr> <tr> <td>001b</td> <td>Divide-by 2</td> <td>101b</td> <td>Reserved</td> </tr> <tr> <td>010b</td> <td>Divide-by 4</td> <td>110b</td> <td>Reserved</td> </tr> <tr> <td>011b</td> <td>Divide-by 8</td> <td>111b</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Description	Bits	Description	000b	Divide-by 1	100b	Divide-by 16	001b	Divide-by 2	101b	Reserved	010b	Divide-by 4	110b	Reserved	011b	Divide-by 8	111b	Reserved																
Bits	Description	Bits	Description																																		
000b	Divide-by 1	100b	Divide-by 16																																		
001b	Divide-by 2	101b	Reserved																																		
010b	Divide-by 4	110b	Reserved																																		
011b	Divide-by 8	111b	Reserved																																		
27:24	<p><b>PowerStepUp.</b> Read-write. Reset: 0000b. Specifies the rate at which blocks of compute unit and NB logic are gated on while the processor transitions from a quiescent state to an active state as part of a power management state transition. There are about 15 steps in this transition for each compute unit and about 5 steps for the NB for the PowerStepDown and PowerStepUp transitions. So the total transition time for a single compute unit is about 15 times the time specified by PowerStepDown and PowerStepUp and the transition time for the NB is about 5 times the time specified by PowerStepDown and PowerStepUp. Use of longer transition times may help reduce voltage transients associated with power state transitions. The bits for PowerStepUp and PowerStepDown are encoded as follows:</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">Bits</th> <th style="text-align: left;">Description</th> <th style="text-align: left;">Bits</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td>0000b</td> <td>Reserved.</td> <td>1000b</td> <td>50 ns</td> </tr> <tr> <td>0001b</td> <td>Reserved.</td> <td>1001b</td> <td>Reserved.</td> </tr> <tr> <td>0010b</td> <td>Reserved.</td> <td>1010b</td> <td>Reserved.</td> </tr> <tr> <td>0011b</td> <td>100 ns</td> <td>1011b</td> <td>Reserved.</td> </tr> <tr> <td>0100b</td> <td>90 ns</td> <td>1100b</td> <td>Reserved.</td> </tr> <tr> <td>0101b</td> <td>80 ns</td> <td>1101b</td> <td>Reserved.</td> </tr> <tr> <td>0110b</td> <td>70 ns</td> <td>1110b</td> <td>Reserved.</td> </tr> <tr> <td>0111b</td> <td>60 ns</td> <td>1111b</td> <td>Reserved.</td> </tr> </tbody> </table> <ul style="list-style-type: none"> <li>• If PowerStepDown or PowerStepUp are programmed to greater than 50 ns, then the value applied to the NB is clipped to 50 ns. The compute unit steps are not clipped.</li> <li>• BIOS: 1000b.</li> </ul>	Bits	Description	Bits	Description	0000b	Reserved.	1000b	50 ns	0001b	Reserved.	1001b	Reserved.	0010b	Reserved.	1010b	Reserved.	0011b	100 ns	1011b	Reserved.	0100b	90 ns	1100b	Reserved.	0101b	80 ns	1101b	Reserved.	0110b	70 ns	1110b	Reserved.	0111b	60 ns	1111b	Reserved.
Bits	Description	Bits	Description																																		
0000b	Reserved.	1000b	50 ns																																		
0001b	Reserved.	1001b	Reserved.																																		
0010b	Reserved.	1010b	Reserved.																																		
0011b	100 ns	1011b	Reserved.																																		
0100b	90 ns	1100b	Reserved.																																		
0101b	80 ns	1101b	Reserved.																																		
0110b	70 ns	1110b	Reserved.																																		
0111b	60 ns	1111b	Reserved.																																		
23:20	<p><b>PowerStepDown.</b> Read-write. Reset: 0000b. BIOS: 1000b. This specifies the rate at which blocks of compute unit and NB logic are gated off while the processor transitions from an active state to a quiescent state as part of a power management state transition.</p>																																				
19:18	Reserved.																																				
17:15	Reserved.																																				
14	<p><b>CacheFlushImmOnAllHalt: cache flush immediate on all halt.</b> Read-write. Reset: 0. 1=Flush the caches immediately when all cores in a package have halted. The following condition must be true in order for the caches to be flushed:</p> <ul style="list-style-type: none"> <li>• <a href="#">D18F4x118/D18F4x11C</a>[CacheFlushEn]=1 for the corresponding C-state action field on all cores.</li> </ul>																																				



13	Reserved.
12	<b>ClkRampHystCtl: clock ramp hysteresis control.</b> Read-write. Reset: 0. Specifies the time base for ClkRampHystSel when (D18F4x128[CoreCstateMode] ? (D18F3x80/D18F3x84[CpuPrbEn]==0) : (D18F4x118/D18F4x11C[CpuPrbEn]==0)). 0=320 ns. 1=1.28 us.
11:8	<p><b>ClkRampHystSel: clock ramp hysteresis select.</b> Read-write. Reset: 0h. BIOS: Fh. When the core(s) are in the stop-grant or halt state and a probe request is received, the core clock may need to be brought up to service the probe.</p> <ul style="list-style-type: none"> <li>If (D18F4x128[CoreCstateMode] ? (D18F3x80/D18F3x84[CpuPrbEn]==0) : (D18F4x118/D18F4x11C[CpuPrbEn]==0)) then this field specifies how long the core clock is left up to service additional probes before being brought back down. Each time a probe request is received, the hysteresis timer is reset such that the period of time specified by this field must expire with no probe request before the core clock is brought back down. The hysteresis time is encoded as (the time base specified by D18F3xD4[ClkRampHystCtl]) * (1 + ClkRampHystSel).</li> <li>If (D18F4x128[CoreCstateMode] ? (D18F3x80/D18F3x84[CpuPrbEn]==1) : (D18F4x118/D18F4x11C[CpuPrbEn]==1)) then this field specifies a fixed amount of time to allow for probes to be serviced after completing the transition of each core. If, for example, two cores enter stop-grant or halt at the same time, then (1) the first core would complete the transition to the low power state, (2) probe traffic would be serviced for the time specified by this field, (3) the second core would complete the transition to the low power state, and (4) probe traffic would be serviced for the time specified by this field (and afterwards, until the next power state transition). For this purpose, values range from 0h=40 ns to Fh=640 ns, encoded as 40 ns * (1 + ClkRampHystSel).</li> </ul>
7:6	Reserved.
5:0	<b>MaxSwPstateCpuCof: maximum software P-state core COF.</b> Read-only. Cold reset: Product-specific. Specifies the maximum CPU COF supported by the processor in a software P-state. The maximum frequency is 100 MHz * MaxSwPstateCpuCof, if MaxSwPstateCpuCof is greater than zero; if MaxSwPstateCpuCof = 00h, then there is no frequency limit. Any attempt to change a software P-state CPU COF to a frequency greater than specified by this field is ignored. See 2.5.3.1.1.1 [Software P-state Numbering] .

### D18F3xD8 Clock Power/Timing Control 1

See 2.5.1.4 [Voltage Transitions].

Bits	Description																				
31:7	Reserved.																				
6:4	<p><b>VSRampSlamTime.</b> Read-write. Cold reset: 000b. BIOS: 100b. Specifies the time the processor waits for voltage transitions to complete before beginning an additional voltage change or a frequency change.</p> <p>Wait time = (VSRampSlamTime / 15mV) * ABS(destination voltage - current voltage).</p> <table border="1" style="margin-left: 20px;"> <thead> <tr> <th><u>Bits</u></th> <th><u>Description</u></th> <th><u>Bits</u></th> <th><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>5.00 us</td> <td>100b</td> <td>2.00 us</td> </tr> <tr> <td>001b</td> <td>3.75 us</td> <td>101b</td> <td>1.50 us</td> </tr> <tr> <td>010b</td> <td>3.00 us</td> <td>110b</td> <td>1.20 us</td> </tr> <tr> <td>011b</td> <td>2.40 us</td> <td>111b</td> <td>1.00 us</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Description</u>	<u>Bits</u>	<u>Description</u>	000b	5.00 us	100b	2.00 us	001b	3.75 us	101b	1.50 us	010b	3.00 us	110b	1.20 us	011b	2.40 us	111b	1.00 us
<u>Bits</u>	<u>Description</u>	<u>Bits</u>	<u>Description</u>																		
000b	5.00 us	100b	2.00 us																		
001b	3.75 us	101b	1.50 us																		
010b	3.00 us	110b	1.20 us																		
011b	2.40 us	111b	1.00 us																		
3:0	Reserved.																				

**D18F3xDC Clock Power/Timing Control 2**

Bits	Description																		
31:30	<b>NbsynPtrAdjPstate[2:1]: NB/core synchronization FIFO pointer adjust P-state[2:1].</b> Read-write. Reset: Product-specific. See NbsynPtrAdj.																		
29:27	<b>NbsynPtrAdjLo: NB/core synchronization FIFO pointer adjust low.</b> Read-write. Cold reset: 000b. BIOS: 101b. See NbsynPtrAdj.																		
26	<b>IgnCpuPrbEn: ignore CPU probe enable.</b> Read-write. Cold reset: 0. BIOS: 0. See <a href="#">D18F3x80/D18F3x84[CpuPrbEn]</a> and <a href="#">D18F4x118/D18F4x11C[CpuPrbEn]</a> .																		
25:19	<b>CacheFlushOnHaltTmr: cache flush on halt timer.</b> Read-write. Cold reset: 00h. IF ( <a href="#">BatteryPower</a> ) THEN BIOS: 4h. ELSE BIOS: Fh. ENDIF. Specifies how long each core needs to stay in a C-state before it flushes its caches. See <a href="#">CacheFlushOnHaltCtl</a> , <a href="#">D18F3x80/D18F3x84[CpuPrbEn]</a> , and <a href="#">D18F4x118/D18F4x11C[CacheFlushTmrSel]</a> . <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00h</td> <td>5.12 us</td> </tr> <tr> <td>7Fh-01h</td> <td>(&lt;CacheFlushOnHaltTmr&gt; * 10.24us) - 5.12us &lt;= Time &lt;= &lt;CacheFlushOnHaltTmr&gt; * 10.24 us</td> </tr> </tbody> </table>	Bits	Description	00h	5.12 us	7Fh-01h	(<CacheFlushOnHaltTmr> * 10.24us) - 5.12us <= Time <= <CacheFlushOnHaltTmr> * 10.24 us												
Bits	Description																		
00h	5.12 us																		
7Fh-01h	(<CacheFlushOnHaltTmr> * 10.24us) - 5.12us <= Time <= <CacheFlushOnHaltTmr> * 10.24 us																		
18:16	<b>CacheFlushOnHaltCtl: cache flush on halt control.</b> Read-write. Reset: 000b. BIOS: 000b. Enables cache flush on halt when ( <a href="#">CacheFlushOnHaltCtl</a> != 0). Specifies what core clock divisor is used after the caches have been flushed. See <a href="#">D18F4x118/D18F4x11C[CacheFlushTmrSel]</a> . <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>Divide-by 1.</td> </tr> <tr> <td>001b</td> <td>Divide-by 2</td> </tr> <tr> <td>010b</td> <td>Divide-by 4</td> </tr> <tr> <td>011b</td> <td>Divide-by 8</td> </tr> <tr> <td>100b</td> <td>Divide-by 16</td> </tr> <tr> <td>101b</td> <td>Reserved</td> </tr> <tr> <td>110b</td> <td>Reserved</td> </tr> <tr> <td>111b</td> <td>Turn off clocks</td> </tr> </tbody> </table> See <a href="#">D18F3x[84:80]</a> and <a href="#">D18F4x11[C:8]</a> for clock divisor specifications that are in effect during a C-state before the caches have been flushed. See <a href="#">2.5.3.2.3.1 [C-state Probes and Cache Flushing]</a> .	Bits	Description	000b	Divide-by 1.	001b	Divide-by 2	010b	Divide-by 4	011b	Divide-by 8	100b	Divide-by 16	101b	Reserved	110b	Reserved	111b	Turn off clocks
Bits	Description																		
000b	Divide-by 1.																		
001b	Divide-by 2																		
010b	Divide-by 4																		
011b	Divide-by 8																		
100b	Divide-by 16																		
101b	Reserved																		
110b	Reserved																		
111b	Turn off clocks																		
15	<b>NbsynPtrAdjPstate[0]: NB/core synchronization FIFO pointer adjust P-state[0].</b> Read-write. Reset: Product-specific. See NbsynPtrAdj.																		

14:12	<p><b>NbsynPtrAdj: NB/core synchronization FIFO pointer adjust.</b> Read-write. Cold reset: 000b. BIOS: 110b.</p> <p>Changes to this field take effect after any of the following events:</p> <ul style="list-style-type: none"> <li>• Warm reset.</li> <li>• At least one core on all compute units perform a P-state transition.</li> <li>• An NB P-state transition.</li> </ul> <p>There is a synchronization FIFO between the NB clock domain and core clock domains. At cold reset, the read pointer and write pointer for each of these FIFOs is positioned conservatively, such that FIFO latency may be greater than is necessary.</p> <p>NbsynPtrAdj and NbsynPtrAdjLo may be used to position the read pointer and write pointer of each FIFO closer to each other such that latency is reduced. Each increment of NbsynPtrAdj and NbsynPtrAdjLo represents one clock cycle of whichever is the slower clock (longer period) between the NB clock and the core clock. NbsynPtrAdj is used when the core P-state is less than or equal to NbsynPtrAdjPstate, otherwise NbsynPtrAdjLo is used.</p> <p>Values less than the recommended value are allowed; values greater than the recommended value are illegal.</p> <table border="1" data-bbox="264 814 1445 955"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>6h-0h</td> <td>Position the read pointer &lt;NbsynPtrAdj, NbsynPtrAdjLo&gt; clock cycles closer to the write pointer.</td> </tr> <tr> <td>7h</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Description	6h-0h	Position the read pointer <NbsynPtrAdj, NbsynPtrAdjLo> clock cycles closer to the write pointer.	7h	Reserved
Bits	Description						
6h-0h	Position the read pointer <NbsynPtrAdj, NbsynPtrAdjLo> clock cycles closer to the write pointer.						
7h	Reserved						
11	Reserved.						
10:8	<p><b>HwpPstateMaxVal: P-state maximum value.</b> Read-write. IF ((D18F3xE8[HtcCapable]==1) &amp;&amp; (D18F3x64[HtcTmpLmt]!=0) &amp;&amp; (D18F3x64[HtcPstateLimit] &gt; HwpPstateMaxVal)) THEN BIOS: D18F3x64[HtcPstateLimit]. ENDIF. Cold reset: specified by the reset state of MSRC001_00[6B:64][PstateEn]; the cold 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 HwpPstateMaxVal=1; if MSRC001_0064 has this bit set and the others do not, then HwpPstateMaxVal=0). This specifies the highest P-state value (lowest performance state) supported by the hardware. This field must not be written to a value less (higher performance) than MSRC001_0071[CurPstateLimit]. See MSRC001_0061[PstateMaxVal]. This field uses hardware P-state numbering. See 2.5.3.1.1.2 [Hardware P-state Numbering].</p>						
7:0	Reserved.						

**D18F3xE4 Thermtrip Status**

Bits	Description
31	<b>SwThermtp: software THERMTRIP.</b> Write-1-only; cleared-by-hardware. 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.
30:6	Reserved.
5	<b>ThermtpEn: THERMTRIP enable.</b> Read-only. Reset: Product-specific. 1=The THERMTRIP state is supported. See 2.10.3.4 [THERMTRIP].

4	Reserved.
3	<b>ThermtpSense: THERMTRIP sense.</b> Read-only. Cold reset: 0. 1=The processor temperature exceeded the THERMTRIP value (regardless as to whether the THERMTRIP state is enabled). This bit is also set when the diagnostic bit SwThermtp = 1.
2	Reserved.
1	<b>Thermtp: THERMTRIP.</b> Read-only. Cold reset: 0. 1=The processor has entered the THERMTRIP state.
0	Reserved.

### D18F3xE8 Northbridge Capabilities

Read-only. Value: Product-specific. Unless otherwise specified, 1=The feature is supported by the processor; 0=The feature is not supported.

Bits	Description
31:29	Reserved.
28	<b>SUCCOR.</b> Read-only. See <a href="#">CPUID Fn8000_0007_EBX[SUCCOR]</a> . Value: 0.
27:26	Reserved.
25	Reserved.
24	<b>MemPstateCap: memory P-state capable.</b>
23:20	Reserved.
19	<b>x2Apic: x2APIC capability.</b> Value: 0.
18:16	Reserved.
15	Reserved.
14	<b>MultVidPlane: multiple VID plane capable.</b> Value: 1.
13:12	Reserved.
11	Reserved.
10	<b>HtcCapable: HTC capable.</b> This affects <a href="#">D18F3x64</a> and <a href="#">D18F3x68</a> .
9	<b>SvmCapable: SVM capable.</b>
8	<b>MctCap: memory controller (on the processor) capable.</b> Value: 1.
7:5	Reserved.
4	<b>ChipKill: chipkill ECC capable.</b>
3	<b>ECC: ECC capable.</b>
2	<b>EightNode: Eight-node multi-processor capable.</b>
1	<b>DualNode: Dual-node multi-processor capable.</b>
0	Reserved.

**D18F3xFC CUID Family/Model/Stepping**

CPUID Fn0000\_0001\_EAX, CPUID Fn8000\_0001\_EAX are an alias of D18F3xFC.

Bits	Description
31:28	Reserved.
27:20	<b>ExtFamily: extended family.</b> Read-only. Value: 07h.
19:16	<b>ExtModel: extended model.</b> Read-only. Value: Product-specific.
15:12	Reserved.
11:8	<b>BaseFamily.</b> Read-only. Reset: Fh.
7:4	<b>BaseModel.</b> Read-only. Value: Product-specific.
3:0	<b>Stepping.</b> Read-only. Value: Product-specific.

**D18F3x138 DCT0 Bad Symbol Identification**

Bits	Description
31:0	Reserved.

**D18F3x13C DCT1 Bad Symbol Identification**

Bits	Description
31:0	Reserved.

**D18F3x140 SRI to XCS Token Count**

Out of cold reset, the processor allocates a minimal number of buffers that is smaller than the default values in the register. BIOS must use D18F0x6C[RIsLnkFullTokCntImm] for the values in the register to take effect. This is necessary even if the values are unchanged from the default values.

D18F3x140, D18F3x144, and D18F3x1[54:48] specify the number of XCS (XBAR command scheduler) entries assigned to each virtual channel within each source port. See 2.8 [Northbridge (NB)].

The default totals are:

Buffer allocation rules:

- The totals of SRI, MCT and the links must not exceed the number of XCS entries.  $XcsSize = 44$ .
- $SUM(D18F3x140[UpReqTok, UpPreqTok, UpRspTok, DnReqTok, DnPreqTok, DnRspTok, IsocReqTok, IsocPreqTok, IsocRspTok, FreeTok]) + SUM(D18F3x144[ProbeTok, RspTok]) + SUM(D18F3x148[ReqTok0, PReqTok0, RspTok0, ProbeTok0, \{FreeTok[3:2], FreeTok[1:0]\}, IsocReqTok0, IsocPreqTok0, IsocRspTok0, ReqTok1, PReqTok1, RspTok1, ProbeTok1, IsocReqTok1, IsocPreqTok1, IsocRspTok1]) + SUM(D18F3x14C[ReqTok0, PReqTok0, RspTok0, ProbeTok0, \{FreeTok[3:2], FreeTok[1:0]\}, IsocReqTok0, IsocPreqTok0, IsocRspTok0, ReqTok1, PReqTok1, RspTok1, ProbeTok1, IsocReqTok1, IsocPreqTok1, IsocRspTok1]) \leq XcsSize$ . See D18F3x1[54:48].

The defaults for [D18F3x140](#) and [D18F3x1\[54:48\]](#) do not allocate any tokens in the isochronous channel. If isochronous flow control mode (IFCM) is enabled ([D18F0x\[E4,C4,A4,84\]\[IsocEn\]](#)), then the XCS token counts must be changed.

- If IFCM is enabled, then [D18F3x140\[IsocReqTok, IsocRspTok\]](#) must each be non-zero. If isochronous posted requests may be generated in the system, then [D18F3x140\[IsocPreqTok\]](#) must also be non-zero.

Bits	Description
31:25	Reserved.
24:20	<b>FreeTok: free tokens.</b> Read-write. Reset: 0Ch. BIOS: Fh. The number of free tokens must always be greater than or equal to 2 to ensure deadlock free operation.
19:18	Reserved.
17:16	<b>IsocRspTok: isochronous response tokens.</b> Read-write. Reset: 0. BIOS: 1.
15:14	<b>IsocPreqTok: isochronous posted request tokens.</b> Read-write. Reset: 0. BIOS: 0.
13:12	<b>IsocReqTok: isochronous request tokens.</b> Read-write. Reset: 0. BIOS: 1.
11:10	<b>DnRspTok: downstream response tokens.</b> Read-write. Reset: 1. BIOS: 1.
9:8	<b>UpRspTok: upstream response tokens.</b> Read-write. Reset: 3. BIOS: 1.
7:6	<b>DnPreqTok: downstream posted request tokens.</b> Read-write. Reset: 1. BIOS: 1.
5:4	<b>UpPreqTok: upstream posted request tokens.</b> Read-write. Reset: 1. BIOS: 2.
3:2	<b>DnReqTok: downstream request tokens.</b> Read-write. Reset: 1. BIOS: 1.
1:0	<b>UpReqTok: upstream request tokens.</b> Read-write. Reset: 3. BIOS: 2.

### D18F3x144 MCT to XCS Token Count

See [D18F3x140](#).

Bits	Description
31:8	Reserved.
7:4	<b>ProbeTok: probe tokens.</b> Read-write. Reset: 7h. BIOS: 4h.
3:0	<b>RspTok: response tokens.</b> Read-write. Reset: 7h. BIOS: 7h.

### D18F3x1[54:48] Link to XCS Token Count

See [D18F3x140](#).

Table 175: [Register Mapping](#) for [D18F3x1\[54:48\]](#)

Register	Function
D18F3x148	ONION Link
D18F3x1[54:4C]	Reserved

Bits	Description
31:30	<b>FreeTok[3:2]: free tokens.</b> Read-write. Cold reset: 00b. BIOS: 0. See FreeTok[1:0].
29	Reserved.
28	<b>IsocRspTok1: isochronous response tokens sublink 1.</b> Read-write. Cold reset: 0. BIOS: 0.
27	Reserved.
26	<b>IsocPreqTok1: isochronous posted request tokens sublink 1.</b> Read-write. Cold reset: 0. BIOS: 0.
25	Reserved.
24	<b>IsocReqTok1: isochronous request tokens sublink 1.</b> Read-write. Cold reset: 0. BIOS: 0.
23:22	<b>ProbeTok1: probe tokens sublink 1.</b> Read-write. Cold reset: 0. BIOS: 0.
21:20	<b>RspTok1: response tokens sublink 1.</b> Read-write. Cold reset: 0. BIOS: 0.
19:18	<b>PReqTok1: posted request tokens sublink 1.</b> Read-write. Cold reset: 0. BIOS: 0.
17:16	<b>ReqTok1: request tokens sublink 1.</b> Read-write. Cold reset: 0. BIOS: 0.
15:14	<b>FreeTok[1:0]: free tokens.</b> Read-write. Cold reset: 00b. FreeTok[3:0] = {FreeTok[3:2], FreeTok[1:0]}. BIOS: 00b.
13:12	<b>IsocRspTok0: isochronous response tokens sublink 0.</b> Read-write. Cold reset: 0. BIOS: 0.
11:10	<b>IsocPreqTok0: isochronous posted request tokens sublink 0.</b> Read-write. Cold reset: 0. BIOS: IF (REG==D18F3x148) THEN 1 ELSE 0 ENDIF. See <a href="#">D18F0x6C</a> [ApplyIsocModeEnNow].

9:8	<b>IsocReqTok0: isochronous request tokens sublink 0.</b> Read-write. Cold reset: 0. BIOS: IF (REG==D18F3x148) THEN 1 ELSE 0 ENDIF.
7:6	<b>ProbeTok0: probe tokens sublink 0.</b> Read-write. Cold reset: 2. BIOS: 0.
5:4	<b>RspTok0: response tokens sublink 0.</b> Read-write. Cold reset: 2. BIOS: IF (REG==D18F3x148) THEN 2 ELSE 0 ENDIF.
3:2	<b>PReqTok0: posted request tokens sublink 0.</b> Read-write. Cold reset: 2. BIOS: IF (REG==D18F3x148) THEN 2 ELSE 0 ENDIF.
1:0	<b>ReqTok0: request tokens sublink 0.</b> Read-write. Cold reset: 2. BIOS: IF (REG==D18F3x148) THEN 2 ELSE 3 ENDIF.

### **D18F3x160 NB Machine Check Misc (DRAM Thresholding) 0 (MC4\_MISC0)**

See 2.14.1.7 [Error Thresholding]. D18F3x160 is associated with the DRAM error type. See MSR0000\_0413.

Bits	Description
31	<b>Valid.</b> Read-only. Reset: 1.
30	<b>CntP: counter present.</b> Read-only. Reset: 1.
29	<b>Locked.</b> Read-only. Reset: 0.
28:24	Reserved.
23:20	<b>LvtOffset: LVT offset.</b> IF (Locked) THEN Read-only. ELSE Read-write. ENDIF. Reset: 0h. BIOS: 1h.
19	<b>CntEn: counter enable.</b> IF (Locked) THEN Read-only. ELSE Read-write. ENDIF. Reset: 0.
18:17	<b>IntType: interrupt type.</b> IF (Locked) THEN Read-only. ELSE Read-write. ENDIF. Cold reset: 0.
16	<b>Ovrflw: overflow.</b> IF (Locked) THEN Read-only; set-by-hardware. ELSE Read-write; set-by-hardware. ENDIF. Cold reset: 0.
15:12	Reserved.
11:0	<b>ErrCnt: error counter.</b> IF (Locked) THEN Read-only; updated-by-hardware. ELSE Read-write; updated-by-hardware. ENDIF. Cold reset: 0.

### **D18F3x168 NB Machine Check Misc (Link Thresholding) 1 (MC4\_MISC1)**

See 2.14.1.7 [Error Thresholding]. D18F3x168 is associated with the link error type. See MSRC000\_0408.

Bits	Description
31	<b>Valid.</b> Read-only. Reset: 1.
30	<b>CntP: counter present.</b> Read-only. Reset: 1.
29	<b>Locked.</b> Read-only. Reset: 0.
28:24	Reserved.
23:20	<b>LvtOffset: LVT offset.</b> IF (Locked) THEN Read-only. ELSE Read-write. ENDIF. Reset: 0h. BIOS: 1h.
19	<b>CntEn: counter enable.</b> IF (Locked) THEN Read-only. ELSE Read-write. ENDIF. Reset: 0.



18:17	<b>IntType: interrupt type.</b> IF (Locked) THEN Read-only. ELSE Read-write. ENDIF. Cold reset: 0.
16	<b>Ovrflw: overflow.</b> IF (Locked) THEN Read-only; set-by-hardware. ELSE Read-write; set-by-hardware. ENDIF. Cold reset: 0.
15:12	Reserved.
11:0	<b>ErrCnt: error counter.</b> IF (Locked) THEN Read-only; updated-by-hardware. ELSE Read-write; updated-by-hardware. ENDIF. Cold reset: 0.

### D18F3x17C Extended Freelist Buffer Count

Out of cold reset, the processor allocates a minimal number of buffers that is smaller than the default values in the register. BIOS must use [D18F0x6C\[RIsLnkFullTokCntImm\]](#) for the values in the register to take effect. This is necessary even if the values are unchanged from the default values.

Bits	Description
31:4	Reserved.
3:0	<b>SPQPrbFreeCBC: XBAR to SRI Probe command buffer freelist.</b> Reset: 8h. Read-write. BIOS: 8h.

### D18F3x180 Extended NB MCA Configuration

Reset: 0000\_0000h. This register is an extension of [D18F3x44 \[MCA NB Configuration\]](#).

Bits	Description
31	Reserved.
30	Reserved.
29	Reserved.
28	<b>SyncFloodOnCC6DramUcErr.</b> Read-write. BIOS: 1. 1=Enable generation of SyncFlood when we hit an Uncorrectable ECC error on C6 restore reads.
27	Reserved.
26	<b>ConvertUnCorToCorErrEn: convert uncorrectable error to correctable error enable.</b> Read-write. 1=The status of uncorrectable errors is changed to appear as correctable errors; <a href="#">MSR0000_0411[UC, PCC]</a> are cleared and a machine check exception will not be raised. For uncorrectable ECC errors, <a href="#">MSR0000_0411[UECC]</a> is cleared and <a href="#">MSR0000_0411[CECC]</a> is set. This field is intended for debug observability.
25	<b>EccSymbolSize: ECC symbol size and code selection.</b> Read-write. BIOS: See <a href="#">2.14.2 [DRAM ECC Considerations]</a> . 0=x4 symbol size and code used. 1=reserved
24	<b>McaLogErrAddrWdtErr: log error address on WDT errors.</b> Read-write. BIOS: 1. 1=When a watchdog timeout error occurs (see <a href="#">MSR0000_0410[WDTRptEn]</a> ), the associated address is logged and <a href="#">MSR0000_0411[AddrV]</a> is set. 0=When a watchdog timeout error occurs, NB state information is saved and <a href="#">MSR0000_0411[AddrV]</a> is cleared. See <a href="#">D18F3x50</a> for details on saved information.

23	Reserved.
22	Reserved.
21	<b>SyncFloodOnCpuLeakErr: sync flood on CPU leak error.</b> Read-write. BIOS: 1. 1=Enable sync flood when one of the cores encounters an uncorrectable error which cannot be contained to the process on the core.
20	Reserved.
19	<b>PwP2pDatErrRmtPropDis: posted write for remote peer-to-peer data error propagation disable.</b> Read-write. 1= A peer-to-peer posted write with a data error is not propagated to the target IO link chain if the target IO link chain is not attached to the local node (the same node as the source IO link chain). Instead, the write is dropped by the host bridge. The state of this field is ignored if SyncFloodOnUsPwDatErr==1 or DatWrErrDeferEn==0.
18	<b>PwP2pDatErrLclPropDis: posted write for local peer-to-peer data error propagation disable.</b> Read-write. 1=A peer-to-peer posted write with a data error is not propagated to the target IO link chain if the target IO link chain is attached to the local node (the same node as the source IO link chain). Instead, the write is dropped by the host bridge. The state of this field is ignored if SyncFloodOnUsPwDatErr==1 or DatWrErrDeferEn==0.
17	<b>SyncFloodOnDeferErrToIO: convert deferred error for an IO link to sync flood enable.</b> Read-write. BIOS: 1. 1=A deferred error which targets an IO link device is turned into a sync flood. <ul style="list-style-type: none"> <li>• When DramErrDeferEn is set and the read response is for a DMA read with a data error, setting SyncFloodOnDeferErrToIO causes a sync flood.</li> <li>• When DatWrErrDeferEn is set and the write is peer-to-peer, setting SyncFloodOnDeferErrToIO causes a sync flood.</li> </ul>
16	<b>DeferDatErrNcHtMcaEn: convert deferred error for an IO link to machine check exception enable.</b> IF (D18F3xE8[SUCCOR]) THEN Read-write. ELSE Read-only. ENDIF. 1=A deferred error which targets an IO link device is turned into a machine check exception. <ul style="list-style-type: none"> <li>• When DramErrDeferEn is set and the read response is for a DMA read with a data error, setting DeferDatErrNcHtMcaEn causes an uncorrected error to be logged and a machine check exception to be generated. An error response is returned to the IO device irrespective of the setting of DeferDatErrNcHtMcaEn.</li> <li>• When DatWrErrDeferEn is set and the write is peer-to-peer, setting DeferDatErrNcHtMcaEn causes an uncorrected error to be logged and a machine check exception to be generated. An error indication is sent to the target IO device irrespective of the setting of DeferDatErrNcHtMcaEn.</li> </ul>
15	Reserved.
14:11	Reserved.
10	Reserved.
9	<b>SyncFloodOnUCNbAry: sync flood on UC NB array error.</b> Read-write. BIOS: 1. 1=Enable sync flood on detection of an UC error in an NB array.
8	<b>SyncFloodOnProtErr: sync flood on protocol error.</b> Read-write. BIOS: 1. 1=Enable sync flood on detection of link protocol error, L3 protocol error, and probe filter protocol error.
7	<b>SyncFloodOnTgtAbortErr.</b> Read-write. BIOS: 1. 1=Enable sync flood on generated or received link responses that indicate target aborts.

6	<b>SyncFloodOnDatErr.</b> Read-write. BIOS: 1. 1=Enable sync flood on generated or received link responses that indicate data error.
5	<b>DisPciCfgCpuMstAbortRsp.</b> Read-write. BIOS: 1. 1=For master abort responses to CPU-initiated configuration accesses, disables MCA error reporting and generation of an error response to the core. 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	<b>ChgMstAbortToNoErr.</b> 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 ChgMstAbortToNoErr and <a href="#">D18F3x44</a> [IoMstAbortDis] are both set, ChgMstAbortToNoErr takes precedence.
3	<b>ChgDatErrToTgtAbort.</b> Read-write. 1=Signal target abort instead of data error in link response packets to IO devices (for Gen1 link compatibility).
2	<b>WDTCntSel[3]: watchdog timer count select bit[3].</b> Read-write. See <a href="#">D18F3x44</a> [WDTCntSel].
1	<b>SyncFloodOnUsPwDatErr: sync flood on upstream posted write data error.</b> Read-write. BIOS: 1. 1=Enable sync flood generation when an upstream posted write data error is detected.
0	<b>McaLogUsPwDatErrEn: MCA log of upstream posted write data error enable.</b> Read-write. BIOS: 1. 1=Enable logging of upstream posted write data errors in MCA (if NB MCA registers are appropriately enabled and configured).

### D18F3x188 NB Configuration 2

Same-for-all.

Bits	Description
31:28	Reserved..
27	<b>DisCpuWrSzDw64ReOrd: disable streaming store reorder.</b> Read-write. Reset: 1. BIOS: 1. 1=Disable reordering of streaming store commands.
26:10	Reserved..
9	<b>DisL3HiPriFreeListAlloc.</b> Read-write. Reset: 0. BIOS: 1. 1=Disables normal SRQ entry scheme which gives higher priority to XBAR.
8:0	Reserved..

### D18F3x190 Downcore Control

Cold reset: 0000\_0000h. See 2.4.4 [Processor Cores and Downcoring] and 2.4.4.1 [Software Downcoring using D18F3x190[DisCore]].

Bits	Description										
31:0	<b>DisCore.</b> Read-write; reset-applied. 0=Core enabled. 1=Core disabled. <table border="0" style="margin-left: 20px;"> <tr> <td style="padding-right: 20px;"><u>Bit</u></td> <td><u>Description</u></td> </tr> <tr> <td>[0]</td> <td>Core 0.</td> </tr> <tr> <td>[2:1]</td> <td>Core &lt;BIT&gt;.</td> </tr> <tr> <td>[3]</td> <td>Core 3.</td> </tr> <tr> <td>[31:4]</td> <td>Reserved.</td> </tr> </table>	<u>Bit</u>	<u>Description</u>	[0]	Core 0.	[2:1]	Core <BIT>.	[3]	Core 3.	[31:4]	Reserved.
<u>Bit</u>	<u>Description</u>										
[0]	Core 0.										
[2:1]	Core <BIT>.										
[3]	Core 3.										
[31:4]	Reserved.										

**D18F3x1A0 Core Interface Buffer Count**

Out of cold reset, the processor allocates a minimal number of buffers that is smaller than the default values in the register. BIOS must use [D18F0x6C\[RIsLnkFullTokCntImm\]](#) for the values in the register to take effect. This is necessary even if the values are unchanged from the default values.

- The following buffer allocations rules must be satisfied:
  - CpuCmdBufCnt >= 2.

Bits	Description								
31	Reserved.								
30:26	<b>NbToCpuPrbLmt.</b> Read-write. Reset: 10h. Maximum number of outstanding probes to the compute-unit. <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>02h-00h</td> <td>Reserved</td> </tr> <tr> <td>0Fh-03h</td> <td>Maximum of &lt;NbToCpuPrbLmt&gt; probes.</td> </tr> <tr> <td>1Fh-10h</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Description	02h-00h	Reserved	0Fh-03h	Maximum of <NbToCpuPrbLmt> probes.	1Fh-10h	Reserved
Bits	Description								
02h-00h	Reserved								
0Fh-03h	Maximum of <NbToCpuPrbLmt> probes.								
1Fh-10h	Reserved								
25:24	Reserved.								
23:20	<b>NbToCpuDatReqLmt.</b> Read-write. Reset: Ch. Octword outstanding per core limit. <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Ch-0h</td> <td>Octword outstanding per core limit</td> </tr> <tr> <td>Fh-Dh</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Description	Ch-0h	Octword outstanding per core limit	Fh-Dh	Reserved		
Bits	Description								
Ch-0h	Octword outstanding per core limit								
Fh-Dh	Reserved								
19	Reserved.								
18:16	<b>CpuToNbFreeBufCnt.</b> Read-write. Reset: 2h. Provides the number of tokens which can be released to each compute unit from the freelist pool. This field can be updated at any time by BIOS and does not require a warm reset to take effect. BIOS: 11b. <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>3h-0h</td> <td>Number of tokens released to each compute unit from the freelist pool.</td> </tr> <tr> <td>7h-4h</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Description	3h-0h	Number of tokens released to each compute unit from the freelist pool.	7h-4h	Reserved		
Bits	Description								
3h-0h	Number of tokens released to each compute unit from the freelist pool.								
7h-4h	Reserved								
15:12	Reserved. Cold reset: 4h.								
11:10	Reserved.								
9:4	Reserved. Cold reset: Product-specific.								
3	Reserved.								
2:0	<b>CpuCmdBufCnt: CPU to SRI command buffer count.</b> Read-write; reset-applied. Each compute unit is allocated the number of buffers specified by this field.  Reset: 2h. BIOS: 1h.								

**D18F3x1CC IBS Control**

Reset: 0000\_0000h. [MSRC001\\_103A](#) is an alias of [D18F3x1CC](#). [D18F3x1CC](#) is programmed by BIOS; The

OS reads the LVT offset from [MSRC001\\_103A](#).

Bits	Description						
31:9	Reserved.						
8	<b>LvtOffsetVal: local vector table offset valid.</b> Read-write. BIOS: 1. 1=The offset in LvtOffset is valid. 0=The offset in LvtOffset is not valid and IBS interrupt generation is disabled.						
7:4	Reserved.						
3:0	<b>LvtOffset: local vector table offset.</b> Read-write. BIOS: 0h. Specifies the address of the IBS LVT entry in the APIC registers. See <a href="#">APIC[530:500]</a> . <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>3h-0h</td> <td>LVT address = &lt;500h + LvtOffset&lt;&lt;4&gt;</td> </tr> <tr> <td>Fh-4h</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Description	3h-0h	LVT address = <500h + LvtOffset<<4>	Fh-4h	Reserved
Bits	Description						
3h-0h	LVT address = <500h + LvtOffset<<4>						
Fh-4h	Reserved						

### D18F3x1FC Product Information Register 1

Bits	Description
31:5	Reserved.
4:1	<b>USB2CompCal[3:0].</b> Value: Product-specific. Specifies USB2.0 common PHY calibration bus value.
0	<b>DcTdpSupport.</b> Value: Product-specific. 1=Dynamic Configurable TDP supported.

### D18F3x200 Performance Mode Control Register

Reset: 0000\_0000h.

Bits	Description
31:8	Reserved.
7:4	<b>EnCpuSkidBufFull.</b> Read-write. Enables optimal use of the CPU skid buffers, in the presence of multiple data movement requests from the same core.
3	<b>EnMcqPrbPickThrottle.</b> Read-write. BIOS: 0. 1=Enabling throttling the MCQ to ensure the bypass path is taken by the probes instead of allocating in to the XCS.
2	<b>EnDctOddToNcLnkDatXfr.</b> Read-write. BIOS: 0. 1=Enables direct transfer of data from odd-numbered DRAM channels (1,3,..) to non-coherent links on the local node.
1	<b>EnDctEvnToNcLnkDatXfr.</b> Read-write. BIOS: 0. 1=Enables direct transfer of data from even-numbered DRAM channels (0,2,..) to non-coherent links on the local node.
0	Reserved.

### D18F3x238 DCT2 Bad Symbol Identification

Bits	Description
31:0	Reserved.

**D18F3x23C DCT3 Bad Symbol Identification**

---

Bits	Description
31:0	Reserved.

**D18F3x2B4 DCT and Fuse Power Gate Control**

---

Cold reset: 0000\_0000h.

Bits	Description
31:27	Reserved.
26:12	Reserved.
11:0	Reserved.

### 3.13 Device 18h Function 4 Configuration Registers

See 3.1 [Register Descriptions and Mnemonics]. See 2.7 [Configuration Space].

#### D18F4x00 Device/Vendor ID

Bits	Description
31:16	<b>DeviceID: device ID.</b> Read-only. Value: 1534h.
15:0	<b>VendorID: vendor ID.</b> Read-only. Value: 1022h.

#### D18F4x04 Status/Command

Bits	Description
31:16	<b>Status.</b> Read-only. Reset: 0000_0000_000X_0000b. Only Status[4] may be set to indicate the existence of a PCI-defined capability block. 0=No supported links are ungangled. 1=At least one link may be ungangled, in which case there is a capability block associated with sublink one of the link in this function.
15:0	<b>Command.</b> Read-only. Value: 0000h.

#### D18F4x08 Class Code/Revision ID

Reset: 0600\_0000h.

Bits	Description
31:8	<b>ClassCode.</b> Read-only. Provides the host bridge class code as defined in the PCI specification.
7:0	<b>RevID: revision ID.</b> Read-only.

#### D18F4x0C Header Type

Reset: 0080\_0000h.

Bits	Description
31:0	<b>HeaderTypeReg.</b> 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

Bits	Description
31:8	Reserved.
7:0	<b>CapPtr: capabilities pointer.</b> Read-only. Value: 00h.

**D18F4x110 Sample and Residency Timers**

Bits	Description
31:21	Reserved.
20:13	<b>MinResTmr: minimum residency timer.</b> IF <a href="#">D18F4x15C</a> [BoostLock] THEN Read-only. ELSE Read-write. ENDIF. Cold reset: Product-specific. Specifies the minimum amount of time required between TDP-initiated P-state transitions. The minimum amount of time is defined as $\text{MinResTmr} * \text{CSampleTimer} * 5.12\mu\text{s}$ .
12	Reserved.
11:0	<b>CSampleTimer.</b> Read-write. Cold reset: 0. BIOS: 14h. Specifies the value that the internal CSampleTimer counter must increment to before expiring. When the internal CSampleTimer counter expires, it is reset to 0. See <a href="#">2.5.9 [Application Power Management (APM)]</a> .

**D18F4x11[C:8] C-state Control**

[D18F4x11\[C:8\]](#) consist of three identical 16-bit registers, one for each C-state Action Field (CAF) associated with an IO address that is read to enter C-states. Refer to [2.5.3.2 \[Core C-states\]](#).

- [D18F4x118](#)[15:0] specifies the actions attempted by the core when software reads from the IO address specified by [MSRC001\\_0073](#)[CstateAddr].
- [D18F4x118](#)[31:16] specifies the actions attempted by the core when software reads from the IO address specified by [MSRC001\\_0073](#)[CstateAddr]+1.
- [D18F4x11C](#)[15:0] specifies the actions attempted by the core when software reads from the IO address specified by [MSRC001\\_0073](#)[CstateAddr]+2.

**D18F4x118 C-state Control 1**

Bits	Description
31:30	Reserved.
29	<b>SelfRefrEarly1.</b> Read-write. Reset: 0. See: SelfRefrEarly0. BIOS: 0.
28	<b>SelfRefr1.</b> Read-write. Reset: 0. See: SelfRefr0. BIOS: 1.
27	<b>NbClkGate1.</b> Read-write. Reset: 0. See: NbClkGate0. BIOS: 1.
26	<b>NbPwrGate1.</b> Read-write. Reset: 0. See: NbPwrGate0.
25	<b>PwrOffEnCstAct1.</b> Read-write; updated-by-SMU. Reset: 0. See: PwrOffEnCstAct0. BIOS: 1.
24	<b>PwrGateEnCstAct1.</b> Read-write. Reset: 0. See: PwrGateEnCstAct0. BIOS: 1.



23:21	<b>ClkDivisorCstAct1</b> . Read-write. Reset: 0. See: ClkDivisorCstAct0. BIOS: 000b.
20	Reserved.
19:18	<b>CacheFlushTmrSelCstAct1</b> . Read-write. Reset: 0. See: CacheFlushTmrSelCstAct0. BIOS: 10b.
17	<b>CacheFlushEnCstAct1</b> . Read-write. Reset: 0. See: CacheFlushEnCstAct0. BIOS: 1.
16	<b>CpuPrbEnCstAct1</b> . Read-write. Reset: 0. See: CpuPrbEnCstAct0. BIOS: 1.
15:14	Reserved.
13	<b>SelfRefrEarly0: allow early self-refresh</b> . Read-write. Reset: 0. BIOS: 0. 1=Allow self-refresh while cores in PC1 or CC1 are waiting for the cache flush timer to expire. 0=Wait for cache flush timer to expire before allowing self-refresh. See <a href="#">2.5.7.2 [DRAM Self-Refresh]</a> and <a href="#">2.5.3.2.3.1 [C-state Probes and Cache Flushing]</a> .
12	<b>SelfRefr0: self-refresh</b> . Read-write. Reset: 0. BIOS: 1. 1=Allow DRAM self-refresh while in NB C-states. 0=Prevent DRAM self-refresh while in NB C-states. NbClkGate0 must be equal to SelfRefr0. See <a href="#">2.5.7.2 [DRAM Self-Refresh]</a> and <a href="#">2.5.4.2 [NB C-states]</a> .
11	<b>NbClkGate0: NB clock-gating</b> . Read-write. Reset: 0. BIOS: 1. 1=Allow clock-gating of the NB. 0=Prevent clock-gating of the NB. NbClkGate0 must be equal to SelfRefr0. See <a href="#">2.5.4.2 [NB C-states]</a> .
10	<b>NbPwrGate0: NB power-gating</b> . Read-write. Reset: 0. 1=Allow power-gating of the NB. 0=Prevent power-gating of the NB. NbPwrGate0 can only be programmed to 1 if NbClkGate0 and SelfRefr0 are programmed to 1. See <a href="#">2.5.4.2 [NB C-states]</a> .
9	<b>PwrOffEnCstAct0: power off enable</b> . Read-write; updated-by-SMU. Reset: 0. BIOS: 1. 1=Package power off enable. CacheFlushEnCstAct0 is required to be set if this bit is set. PwrGateEnCstAct0 is required to be set if this bit is set. See <a href="#">2.5.3.2.3.4 [Package C6 (PC6) State]</a> .
8	<b>PwrGateEnCstAct0: power gate enable</b> . Read-write. Reset: 0. BIOS: 1. 1=Core power gating is enabled. CacheFlushEnCstAct0 is required to be set if this bit is set. See <a href="#">2.5.3.2.3.3 [Core C6 (CC6) State]</a> .

7:5	<p><b>ClkDivisorCstAct0: clock divisor.</b> Read-write. Reset: 0. BIOS: 000b.</p> <p>Specifies the core clock frequency while in the low-power state before the caches are flushed. This divisor is relative to the current FID frequency, or:</p> <ul style="list-style-type: none"> <li>100 MHz * (10h + MSRC001_00[6B:64][CpuFid]) of the current P-state specified by MSRC001_0063[CurPstate].</li> </ul> <p>If MSRC001_00[6B:64][CpuDid] of the current P-state indicates a divisor that is deeper than specified by this field, then no frequency change is made when entering the low-power state associated with this register.</p> <table border="1" data-bbox="263 506 998 674"> <thead> <tr> <th>Bits</th> <th>Description</th> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>/1</td> <td>100b</td> <td>/16</td> </tr> <tr> <td>001b</td> <td>/2</td> <td>101b</td> <td>/128</td> </tr> <tr> <td>010b</td> <td>/4</td> <td>110b</td> <td>/512</td> </tr> <tr> <td>011b</td> <td>/8</td> <td>111b</td> <td>Turn off clocks.</td> </tr> </tbody> </table> <p>See CacheFlushTmrSelCstAct0.</p>	Bits	Description	Bits	Description	000b	/1	100b	/16	001b	/2	101b	/128	010b	/4	110b	/512	011b	/8	111b	Turn off clocks.
Bits	Description	Bits	Description																		
000b	/1	100b	/16																		
001b	/2	101b	/128																		
010b	/4	110b	/512																		
011b	/8	111b	Turn off clocks.																		
4	Reserved.																				
3:2	<p><b>CacheFlushTmrSelCstAct0: cache flush timer select.</b> Read-write. Reset: 0. BIOS: 10b. Specifies the timer to use for cache flush.</p> <table border="1" data-bbox="263 842 836 1010"> <thead> <tr> <th>Bits</th> <th>Cache flush timer</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>0 us</td> </tr> <tr> <td>01b</td> <td>D18F3xDC[CacheFlushOnHaltTmr]</td> </tr> <tr> <td>10b</td> <td>D18F4x128[CacheFlushTmr]</td> </tr> <tr> <td>11b</td> <td>Reserved</td> </tr> </tbody> </table> <p>Each core has one timer.</p> <p>D18F3xDC[CacheFlushOnHaltCtl] specifies the core clock divisor to use after the caches are flushed. Writing values greater than 10b result in 10b. See CacheFlushEnCstAct0 and CpuPrbEnCstAct0.</p>	Bits	Cache flush timer	00b	0 us	01b	D18F3xDC[CacheFlushOnHaltTmr]	10b	D18F4x128[CacheFlushTmr]	11b	Reserved										
Bits	Cache flush timer																				
00b	0 us																				
01b	D18F3xDC[CacheFlushOnHaltTmr]																				
10b	D18F4x128[CacheFlushTmr]																				
11b	Reserved																				
1	<p><b>CacheFlushEnCstAct0: cache flush enable.</b> Read-write. Reset: 0. BIOS: 1. 1=Cache flush enable. The cache flush timer starts counting when the C-state is entered. See CacheFlushTmrSelCstAct0 and 2.5.3.2.3.1 [C-state Probes and Cache Flushing].</p> <p>PwrGateEnCstAct0 is required to be set if this bit is set.</p>																				
0	<p><b>CpuPrbEnCstAct0: core direct probe enable.</b> Read-write. Reset: 0. BIOS: 1. Specifies how probes are handled while in the low-power state. 0=When the probe request comes into the NB, the core clock is brought up to the COF (based on the current P-state), all outstanding probes are completed, the core waits for a hysteresis time based on D18F3xD4[ClkRampHystSel], and then the core clock is brought down to the frequency specified by ClkDivisorCstAct0. 1=The core clock does not change frequency; the probe is handled at the frequency specified by ClkDivisorCstAct0; this may only be set if:</p> <ul style="list-style-type: none"> <li>ClkDivisorCstAct0 specifies a divide-by 1, 2, 4, 8, or 16 and NbCof &lt;= 3.2 GHz</li> <li>ClkDivisorCstAct0 specifies a divide-by 1, 2, 4, or 8 and NbCof &gt;= 3.4 GHz</li> </ul> <p>This bit also specifies functionality of the timer used for cache flushing. See CacheFlushTmrSelCstAct0.</p> <ul style="list-style-type: none"> <li>If CpuPrbEnCstAct0=0 and D18F3xDC[IgnCpuPrbEn]=0, only the time when the core is in a non-C0 state and has its clocks ramped up to service probes is counted.</li> <li>If CpuPrbEnCstAct0=1 or D18F3xDC[IgnCpuPrbEn]=1, all of the time the core is in a non-C0 state is counted.</li> </ul>																				

**D18F4x11C C-state Control 2**

Reset: 0000\_0000h. Read-write.

Bits	Description
31:14	Reserved.
13	<b>SelfRefrEarly2</b> . See: <a href="#">D18F4x118[SelfRefrEarly0]</a> .
12	<b>SelfRefr2</b> . See: <a href="#">D18F4x118[SelfRefr0]</a> .
11	<b>NbClkGate2</b> . See: <a href="#">D18F4x118[NbClkGate0]</a> .
10	<b>NbPwrGate2</b> . See: <a href="#">D18F4x118[NbPwrGate0]</a> .
9	<b>PwrOffEnCstAct2</b> . See: <a href="#">D18F4x118[PwrOffEnCstAct0]</a> .
8	<b>PwrGateEnCstAct2</b> . See: <a href="#">D18F4x118[PwrGateEnCstAct0]</a> .
7:5	<b>ClkDivisorCstAct2</b> . See: <a href="#">D18F4x118[ClkDivisorCstAct0]</a> .
4	Reserved.
3:2	<b>CacheFlushTmrSelCstAct2</b> . See: <a href="#">D18F4x118[CacheFlushTmrSelCstAct0]</a> .
1	<b>CacheFlushEnCstAct2</b> . See: <a href="#">D18F4x118[CacheFlushEnCstAct0]</a> .
0	<b>CpuPrbEnCstAct2</b> . See: <a href="#">D18F4x118[CpuPrbEnCstAct0]</a> .

**D18F4x124 C-state Interrupt Control**

Bits	Description
31:0	Reserved.

**D18F4x128 C-state Policy Control 1**

Reset: 0080\_0000h.

Bits	Description										
31	<b>CstateMsgDis: C-state messaging</b> disable. Read-write. Specifies whether any messages are sent to the FCH when a core enters or exits a C-state. 0=Messages are sent. 1=Messages are not sent. See <a href="#">2.5.3.2.4.1 [FCH Messaging]</a> .										
30:25	Reserved.										
24:23	<b>CacheFlushSucMonMispredictAct: cache flush success monitor mispredict action</b> . Read-write. BIOS: 01b. Specifies the cache flush success monitor decrement when non-C0 residency is shorter than duration specified by CacheFlushSucMonTmrSel. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>reset counter to zero</td> </tr> <tr> <td>01b</td> <td>decrement by 1</td> </tr> <tr> <td>10b</td> <td>decrement by 2</td> </tr> <tr> <td>11b</td> <td>decrement by 2</td> </tr> </tbody> </table>	Bits	Description	00b	reset counter to zero	01b	decrement by 1	10b	decrement by 2	11b	decrement by 2
Bits	Description										
00b	reset counter to zero										
01b	decrement by 1										
10b	decrement by 2										
11b	decrement by 2										

22:21	<p><b>CacheFlushSucMonTmrSel: cache flush success monitor timer select.</b> Read-write.                  BIOS: 01b.                  Specifies the non-C0 duration used to increment the cache flush success monitor.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Duration</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Use cache flush timer specified by <a href="#">D18F4x11[C:8]</a></td> </tr> <tr> <td>01b</td> <td><a href="#">D18F3xDC</a>[CacheFlushOnHaltTmr]</td> </tr> <tr> <td>10b</td> <td><a href="#">D18F4x128</a>[CacheFlushTmr]</td> </tr> <tr> <td>11b</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Duration	00b	Use cache flush timer specified by <a href="#">D18F4x11[C:8]</a>	01b	<a href="#">D18F3xDC</a> [CacheFlushOnHaltTmr]	10b	<a href="#">D18F4x128</a> [CacheFlushTmr]	11b	Reserved
Bits	Duration										
00b	Use cache flush timer specified by <a href="#">D18F4x11[C:8]</a>										
01b	<a href="#">D18F3xDC</a> [CacheFlushOnHaltTmr]										
10b	<a href="#">D18F4x128</a> [CacheFlushTmr]										
11b	Reserved										
20:18	<p><b>CacheFlushSucMonThreshold: cache flush success monitor threshold.</b> Read-write.                  BIOS: 100b.                  Flush the caches immediately if cache flushing is enabled and the cache flush success monitor count == CacheFlushSucMonThreshold. A value of 0 disables the cache flush success monitor. See <a href="#">D18F4x118/D18F4x11C</a>[CacheFlushEn].</p>										
17:12	Reserved.										
11:5	<p><b>CacheFlushTmr: cache flush timer.</b> Read-write.                  IF (<a href="#">BatteryPower</a>) THEN BIOS: 3Ch. ELSE BIOS: 7Fh. ENDIF.                  Specifies how long each core needs to stay in a C-state before it flushes its caches. See <a href="#">D18F4x118/D18F4x11C</a>[CacheFlushTmrSel].</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00h</td> <td>&lt;= 5.12 us</td> </tr> <tr> <td>7Fh-01h</td> <td>(&lt;CacheFlushTmr&gt; * 10.24us) - 5.12us &lt;= Time &lt;= &lt;CacheFlushTmr&gt; * 10.24 us</td> </tr> </tbody> </table>	Bits	Description	00h	<= 5.12 us	7Fh-01h	(<CacheFlushTmr> * 10.24us) - 5.12us <= Time <= <CacheFlushTmr> * 10.24 us				
Bits	Description										
00h	<= 5.12 us										
7Fh-01h	(<CacheFlushTmr> * 10.24us) - 5.12us <= Time <= <CacheFlushTmr> * 10.24 us										
4:2	<p><b>HaltCstateIndex.</b> Read-write. BIOS: 0. Specifies the IO-based C-state that is invoked by a HLT instruction.</p>										
1	Reserved.										
0	Reserved.										

**D18F4x13C SMU P-state Control**

Reset: 0000\_0000h. Read-only, updated-by-SMU.

Bits	Description
31:4	Reserved.
3:1	<p><b>SmuPstateLimit.</b> Specifies the highest-performance P-state (lowest value) allowed. SmuPstateLimit is always bounded by <a href="#">MSRC001_0061</a>[PstateMaxVal]. This field uses hardware P-state numbering. See <a href="#">MSRC001_0071</a>[CurPstateLimit] and <a href="#">2.5.3.1.1.2</a> [<a href="#">Hardware P-state Numbering</a>].</p>
0	<b>SmuPstateLimitEn.</b>

**D18F4x15C Core Performance Boost Control**

Bits	Description										
31	<p><b>BoostLock.</b> Read-only. Reset: Product-specific. Specifies whether the following registers are Read-write, read-only, or have special requirements related to writability. See individual register definitions for details.</p> <ul style="list-style-type: none"> <li>• MSRC001_00[6B:64][CpuFid, CpuDid, CpuVid].</li> <li>• D18F4x110[MinResTmr]</li> <li>• D18F4x15C[NumBoostStates].</li> <li>• D18F4x16C[CstateCnt, CstateBoost].</li> <li>• D18F4x250[NodeTdpLimit].</li> </ul>										
30:9	Reserved.										
8	<p><b>CstatePowerEn: C-state power enable.</b> If D18F2x1B4[SmuCfgLock] THEN Read-only; updated-by-hardware. ELSE Read-write. ENDIF Reset: 0. BIOS: 1.</p>										
7	<p><b>ApmMasterEn: APM master enable.</b> If D18F2x1B4[SmuCfgLock] THEN Read-only; updated-by-hardware. ELSE Read-write. ENDIF Reset: 0. BIOS: IF(D18F4x15C[NumBoostStates]==0) THEN 0. ELSE 1. ENDIF. 1=Enables the ability to turn on features associated with APM when used in conjunction with the individual feature enable bits. See 2.5.9 [Application Power Management (APM)].</p>										
6:5	Reserved.										
4:2	<p><b>NumBoostStates: number of boosted states.</b> IF (D18F4x15C[BoostLock]   ApmMasterEn   D18F2x1B4[SmuCfgLock]) THEN Read-only. ELSE Read-write. ENDIF. Reset: Product-specific. Specifies the number of P-states that are considered boosted P-states. See 2.5.9 [Application Power Management (APM)].</p>										
1:0	<p><b>BoostSrc: boost source.</b> If D18F2x1B4[SmuCfgLock] THEN Read-only; updated-by-hardware. ELSE Read-write. ENDIF Reset: 0. BIOS: 2.5.3.1.6. Specifies whether CPB is enabled or disabled.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Boosting disabled</td> </tr> <tr> <td>01b</td> <td>Boosting enabled</td> </tr> <tr> <td>10b</td> <td>Reserved</td> </tr> <tr> <td>11b</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Description	00b	Boosting disabled	01b	Boosting enabled	10b	Reserved	11b	Reserved
Bits	Description										
00b	Boosting disabled										
01b	Boosting enabled										
10b	Reserved										
11b	Reserved										

**D18F4x164 Fixed Errata**

Bits	Description
31:0	<p><b>FixedErrata.</b> Value: Product-specific. See the Revision Guide for the definition of this field. See 1.2 [Reference Documents].</p>

**D18F4x16C APM TDP Control**

Bits	Description						
31:15	Reserved.						
14	<b>CacUpC1.</b> IF <a href="#">D18F4x15C</a> [BoostLock] THEN Read-only. ELSE Read-write. ENDIF. Reset: Product-specific. BIOS: 0. 1=Cac interface is up on C1 (non XC6) state. 0=Cac interface is down and Cstate scalers are used in place of Cac reads.						
13	<b>CstateCores.</b> IF <a href="#">D18F4x15C</a> [BoostLock] THEN Read-only. ELSE Read-write. ENDIF. Reset: Product-specific. Specifies how CstateCnt determines Cstate boost conditions. <table border="1"> <thead> <tr> <th>Bit</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>CstateCnt specifies the number of cores.</td> </tr> <tr> <td>1h</td> <td>CstateCnt specifies the number of compute units.</td> </tr> </tbody> </table>	Bit	Description	0h	CstateCnt specifies the number of cores.	1h	CstateCnt specifies the number of compute units.
Bit	Description						
0h	CstateCnt specifies the number of cores.						
1h	CstateCnt specifies the number of compute units.						
12	Reserved.						
11:9	<b>CstateCnt: C-state count.</b> IF <a href="#">D18F4x15C</a> [BoostLock] THEN Read-only. ELSE Read-write. ENDIF. Reset: Product-specific. Specifies the number of cores or compute units (see CstateCores) that must be in CC6 before a transition can occur to a boosted P-state that is higher performance than the P-state specified by CstateBoost. A value of 0 disables access to P-states above CstateBoost.						
8:6	<b>CstateBoost.</b> Read-write. Reset: Product-specific. Specifies the P-state which requires the number of cores or compute units (see CstateCores) specified in CstateCnt to be in CC6 before a transition to a higher performance (lower numbered) boosted P-state is allowed. CstateBoost must be less than or equal to <a href="#">D18F4x15C</a> [NumBoostStates] otherwise undefined behavior results. If <a href="#">D18F4x15C</a> [BoostLock]=1, CstateBoost can only be written with values that are greater than or equal to the reset value. Attempts to write values less than the reset value are ignored. A value of 0 indicates that the C-state boost feature is not supported. This field uses hardware P-state numbering. See <a href="#">2.5.3.1.1.2 [Hardware P-state Numbering]</a> .						
5	<b>ApmTdpLimitSts: APM TDP limit status.</b> Read; set-by-hardware; write-1-to-clear. Reset: 0. This bit is set by hardware when <a href="#">D18F5xE8</a> [ApmTdpLimit] changes.						
4	<b>ApmTdpLimitIntEn: APM TDP limit interrupt enable.</b> Read-write. Reset: 0. BIOS: 1. 1=Enables the generation of an interrupt using <a href="#">APIC330</a> of each core when <a href="#">D18F5xE8</a> [ApmTdpLimit] changes.						
3	<b>TdpLimitDis.</b> IF <a href="#">D18F4x15C</a> [BoostLock] THEN Read-only. ELSE Read-write. ENDIF. IF ( <a href="#">D0F0xBC_xC010408C</a> [2]==1) THEN BIOS:1. ENDIF. Reset: 0. 1=Disables TDP limit checking and allows the processor to transition to higher performance P-states.						
2:0	Reserved.						

**D18F4x1C0 Node Cac Register 1**

Bits	Description
31:12	Reserved.
11:0	<b>NodeCacLatest.</b> Read-only, updated-by-hardware. Reset: 0. Specifies the sum of all instantaneous power credits on each compute unit. NodeCacLatest is reset to 0 when <a href="#">D18F4x15C</a> [ApmMasterEn]=0.

**D18F4x250 TDP Limit 8**

Bits	Description
31	Reserved.
30:28	<b>TdpLimitPstate.</b> Read-write. Reset: 0. Specifies the highest performance P-state that has a power consumption less than or equal to the TDP limit. This field is programmed by BIOS and uses software P-state numbering. See <a href="#">2.5.3.1.1.1 [Software P-state Numbering]</a> .
27:12	Reserved.
11:0	<b>NodeTdpLimit.</b> Read-write; <b>Same-for-all.</b> Reset: Product-specific. Specifies the maximum allowed sum of TDPs from all cores on a node. If the consumed power exceeds the NodeTdpLimit, a P-state limit is applied to all cores on the processor to reduce the power consumption so that it remains within the TDP limit. If <a href="#">D18F4x15C[BoostLock]=1</a> , NodeTdpLimit can only be written with values that are less than or equal to the reset value. Attempts to write an invalid value are ignored. See <a href="#">2.5.9.2 [TDP Limiting]</a> .

### 3.14 Device 18h Function 5 Configuration Registers

See 3.1 [Register Descriptions and Mnemonics]. See 2.7 [Configuration Space].

#### D18F5x00 Device/Vendor ID

Bits	Description
31:16	<b>DeviceID: device ID.</b> Read-only. Value: 1535h.
15:0	<b>VendorID: vendor ID.</b> Read-only. Value: 1022h.

#### D18F5x04 Status/Command

Bits	Description
31:16	<b>Status.</b> Read-only. Value: 0000h.
15:0	<b>Command.</b> Read-only. Value: 0000h.

#### D18F5x08 Class Code/Revision ID

Bits	Description
31:8	<b>ClassCode.</b> Read-only. Value: 060000h. Provides the host bridge class code as defined in the PCI specification.
7:0	<b>RevID: revision ID.</b> Read-only. Value: 00h.

#### D18F5x0C Header Type

Bits	Description
31:0	<b>HeaderTypeReg.</b> Read-only. Reset: 0080_0000h. These bits are fixed at their default values. The header type field indicates that there are not multiple functions present in this device.

#### D18F5x34 Capabilities Pointer

Bits	Description
31:8	Reserved.
7:0	<b>CapPtr: capabilities pointer.</b> Read-only. Value: 00h.

#### D18F5x[70,60,50,40] Northbridge Performance Event Select Low

Bits	Description
31:0	<b>MSRC001_024[6,4,2,0][31:0]</b> is an alias of <b>D18F5x[70,60,50,40]</b> .



**D18F5x[74,64,54,44] Northbridge Performance Event Select High**

Bits	Description
31:0	MSRC001_024[6,4,2,0][63:32] is an alias of D18F5x[74,64,54,44].

**D18F5x[78,68,58,48] Northbridge Performance Event Counter Low**

Bits	Description
31:0	MSRC001_024[7,5,3,1][31:0] is an alias of D18F5x[78,68,58,48].

**D18F5x[7C,6C,5C,4C] Northbridge Performance Event Counter High**

Bits	Description
31:0	MSRC001_024[7,5,3,1][63:32] is an alias of D18F5x[7C,6C,5C,4C].

**D18F5x80 Compute Unit Status 1**

See 2.4.4 [Processor Cores and Downcoring].

Software associates core ID to the cores of the compute units according to the following table. All combinations not listed are reserved.

**Table 176: D18F5x80[Enabled, DualCore, TripleCore, QuadCore] Definition**

Enabled	DualCore	TripleCore	QuadCore	Definition
1h	xh	xh	1h	1 L2 complex is enabled; four cores of the L2 complex are enabled.
1h	xh	1h	0h	1 L2 complex is enabled; three cores of the L2 complex are enabled.
1h	1h	0h	0h	1 L2 complex is enabled; two cores of the L2 complex are enabled.
1h	0h	0h	0h	1 L2 complex is enabled; one core of the L2 complex are enabled.

Bits	Description										
31:28	Reserved.										
27:24	<b>QuadCore: four cores of a L2 complex are enabled.</b> Read-only. Reset: Product-specific. 1=Four cores of a L2 complex are enabled. See Table 176 [D18F5x80[Enabled, DualCore, TripleCore, QuadCore] Definition]. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Bit</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>[0]</td> <td>L2 complex 0</td> </tr> <tr> <td>[1]</td> <td>Reserved</td> </tr> <tr> <td>[2]</td> <td>Reserved</td> </tr> <tr> <td>[3]</td> <td>Reserved</td> </tr> </tbody> </table>	Bit	Description	[0]	L2 complex 0	[1]	Reserved	[2]	Reserved	[3]	Reserved
Bit	Description										
[0]	L2 complex 0										
[1]	Reserved										
[2]	Reserved										
[3]	Reserved										
23:20	Reserved.										

19:16	<p><b>DualCore: two cores of a L2 complex are enabled.</b> Read-only. Reset: Product-specific. 1=Both cores of a L2 complex are enabled. See <a href="#">Table 176 [D18F5x80[Enabled, DualCore, TripleCore, QuadCore] Definition]</a>.</p> <table border="1"> <thead> <tr> <th>Bit</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>[0]</td> <td>L2 complex 0</td> </tr> <tr> <td>[1]</td> <td>Reserved</td> </tr> <tr> <td>[2]</td> <td>Reserved</td> </tr> <tr> <td>[3]</td> <td>Reserved</td> </tr> </tbody> </table>	Bit	Description	[0]	L2 complex 0	[1]	Reserved	[2]	Reserved	[3]	Reserved
Bit	Description										
[0]	L2 complex 0										
[1]	Reserved										
[2]	Reserved										
[3]	Reserved										
15:12	Reserved.										
11:8	<p><b>TripleCore: three cores of a L2 complex are enabled.</b> Read-only. Reset: Product-specific. 1=Three cores of a L2 complex are enabled. See <a href="#">Table 176 [D18F5x80[Enabled, DualCore, TripleCore, QuadCore] Definition]</a>.</p> <table border="1"> <thead> <tr> <th>Bit</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>[0]</td> <td>L2 complex 0</td> </tr> <tr> <td>[1]</td> <td>Reserved</td> </tr> <tr> <td>[2]</td> <td>Reserved</td> </tr> <tr> <td>[3]</td> <td>Reserved</td> </tr> </tbody> </table>	Bit	Description	[0]	L2 complex 0	[1]	Reserved	[2]	Reserved	[3]	Reserved
Bit	Description										
[0]	L2 complex 0										
[1]	Reserved										
[2]	Reserved										
[3]	Reserved										
7:4	Reserved.										
3:0	<p><b>Enabled: at least one core of a L2 complex is enabled.</b> Read-only. Reset: Product-specific. 1=At least one core is enabled in a L2 complex. See <a href="#">Table 176 [D18F5x80[Enabled, DualCore, TripleCore, QuadCore] Definition]</a>.</p> <table border="1"> <thead> <tr> <th>Bit</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>[0]</td> <td>L2 complex 0</td> </tr> <tr> <td>[1]</td> <td>Reserved</td> </tr> <tr> <td>[2]</td> <td>Reserved</td> </tr> <tr> <td>[3]</td> <td>Reserved</td> </tr> </tbody> </table>	Bit	Description	[0]	L2 complex 0	[1]	Reserved	[2]	Reserved	[3]	Reserved
Bit	Description										
[0]	L2 complex 0										
[1]	Reserved										
[2]	Reserved										
[3]	Reserved										

### D18F5x84 Northbridge Capabilities 2

Unless otherwise specified, 1=The feature is supported by the processor; 0=The feature is not supported.

Bits	Description						
31:29	Reserved.						
28:24	<p><b>DdrMaxRateEnf: enforced maximum DDR rate.</b> Read-only. Value: Product-specific. See: DdrMaxRate. Specifies the maximum DRAM data rate that the processor is designed to support. Writes to <a href="#">D18F2x94_dct[0][MemClkFreq]</a> that specify a frequency greater than specified by DdrMaxRateEnf will result in the <a href="#">D18F2x94_dct[0][MemClkFreq]</a> being set to DdrMaxRateEnf.</p>						
23:21	Reserved.						
20:16	<p><b>DdrMaxRate: maximum DDR rate.</b> Read-only. Value: Product-specific. Specifies the maximum DRAM data rate that the processor is designed to support. DdrMaxRate is defined by <a href="#">Table 125 [Valid Values for Memory Clock Frequency Value Definition]</a>; except that 00h is defined as no limit. See <a href="#">D18F2x94_dct[0][MemClkFreq]</a>, and DdrMaxRateEnf.</p>						
15:12	<p><b>DctEn[3:0]: DCT[3:0] enabled.</b> Read-only. Value: Product-specific. Specifies which DCT controllers are enabled. 1=Enabled. 0=Disabled.</p> <table border="1"> <thead> <tr> <th>Bit</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>[0]</td> <td>DCT 0</td> </tr> <tr> <td>[3:1]</td> <td>Reserved</td> </tr> </tbody> </table>	Bit	Description	[0]	DCT 0	[3:1]	Reserved
Bit	Description						
[0]	DCT 0						
[3:1]	Reserved						

11:8	Reserved.
7:0	<b>CmpCap: CMP capable.</b> Read-only. Value: Product-specific. Number of cores on the node is CmpCap+1. CmpCap does not reflect cores disabled by <a href="#">D18F3x190</a> [DisCore].

#### D18F5x88 NB Configuration 4 (NB\_CFG4)

Bits	Description
31:25	Reserved.
24	<b>DisHbNpReqBusLock.</b> Read-write. Reset: 0. BIOS: 1. 0=While bus locks are in progress, all non-posted commands from I/O, including atomics, are blocked until the core has completed the locked transaction and releases the bus. 1=All non-posted commands except atomics do not honor bus locks and are allowed to proceed. This bit may be set to achieve better DMA performance in the presence of bus locks.
23:19	Reserved.
18	<b>EnCstateBoostBlockCC6Exit.</b> Read-write. Reset: 0. If (EnCstateBoostBlockCC6Exit==1), cores cannot exit CC6 until VDD is less than or equal to the voltage of the P-state indexed by <a href="#">D18F4x16C</a> [CstateBoost].
17:15	Reserved.
14	<b>DisHldRegRdRspChk.</b> Read-write. Reset: 0. 1=Disable primary holding register CPU or I/O read response checks.
13:1	Reserved.
0	<b>CC6PstateWakeUpDis.</b> Read-write. Reset: 0. BIOS: 1. 1=Disable waking up cores in CC6 for P-state changes. 0=Wakeup cores in CC6 for a P-state change to the PopDownPstate, and then return the core to CC6.

#### D18F5x8C NB Configuration 5 (NB\_CFG5)

Bits	Description
31:16	Reserved.
15	<b>EnSrqAllocGt31.</b> Read-write. Reset: 0b. BIOS: 1. 1=Enables allocation of SRA entries to above the lower 32 entries.
14:0	Reserved.

#### D18F5xE0 Processor TDP Running Average

Bits	Description
31:4	Reserved.
3:0	<b>RunAvgRange: running average range.</b> Read-write; <b>Same-for-all.</b> Reset: 0. BIOS: 1h. Specifies the interval over which the processor averages power consumption estimates from the cores for boosting. Time interval = $2^{(\text{RunAvgRange} + 1)} * \text{FreeRunSampleTimer}$ rate. A value of 0 disables the TDP running average accumulator capture function. See <a href="#">2.5.9 [Application Power Management (APM)]</a> .

**D18F5xE8 TDP Limit 3**

Bits	Description
31:29	Reserved.
28:16	<b>ApmTdpLimit.</b> Read-only; updated-by-hardware. Value: <a href="#">D18F4x250[NodeTdpLimit]</a> . If the consumed node power exceeds the ApmTdpLimit on an single node processor or the ApmTdpLimit/2 on a multi-node processor, a P-state limit is applied to all cores on all nodes to reduce the power consumption to remain within the TDP limit. See <a href="#">2.5.9.2 [TDP Limiting]</a> .
15:10	Reserved.
9:0	<b>Tdp2Watt.</b> Read-only. Value: Product-specific. Specifies in watts/TDP units the conversion factor for converting TDP units to watts. Tdp2Watt is a fixed point integer with 10 bits to the right of the decimal point and 0 bits to the left of the decimal point.

**D18F5xEC Load Step Throttle Control**

Bits	Description
31:0	Reserved.

**D18F5x128 Clock Power/Timing Control 3**

Bits	Description												
31	Reserved.												
30	<b>NbFidChgCpuOpEn.</b> Read-write. Reset: 0b.												
29:28	Reserved.												
27	<b>SprSaveRestoreEn.</b> Read-write. Cold Reset:0. Enables SPR save/restore for non-retention NB power gating.												
26:23	Reserved.												
22	<b>NbPllPwrDwnRegEn: NB PLL power down.</b> Read-write. Cold reset: Product-specific. 1=The NB PLL is powered down when the NB is power gated and DRAM is placed into self-refresh (see <a href="#">2.5.4.2 [NB C-states]</a> ). 0=The NB PLL is not powered down during NB C-states.												
21	<b>PC6Vid[7].</b> Read-write. Cold reset: Product-specific. See PC6Vid[6:0].												
20:16	Reserved.												
15	<b>CC6PwrDwnRegEn: CC6 power down regulator enable.</b> Read-write. Cold reset: Product-specific. 1=Power down the VDDA regulator on CC6 entry. See PllRegTime.												
14	<b>PC6PwrDwnRegEn: PC6 power down regulator enable.</b> Read-write. Cold reset: Product-specific. 1=Power down the VDDA regulator on PC6 entry. See PllRegTime.												
13:12	<b>PwrGateTmr: power gate timer.</b> Read-write. Cold reset: 01b. BIOS: 00b. Specifies the minimum delay time required from the power gating or ungating of one Core to the power gating or ungating of the same Core or another Core. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Bits</th> <th>Description</th> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>500 ns</td> <td>10b</td> <td>Reserved.</td> </tr> <tr> <td>01b</td> <td>1 us</td> <td>11b</td> <td>Reserved.</td> </tr> </tbody> </table>	Bits	Description	Bits	Description	00b	500 ns	10b	Reserved.	01b	1 us	11b	Reserved.
Bits	Description	Bits	Description										
00b	500 ns	10b	Reserved.										
01b	1 us	11b	Reserved.										

11:10	<p><b>PllVddOutUpTime.</b> Read-write. Reset: 0. The VDD regulator may be powered down when the processor transitions to PC6. If the regulator is powered down, this field specifies the time required to initialize the core PLL logic once the regulator is powered back up.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>100 ns</td> <td>10b</td> <td>400 ns</td> </tr> <tr> <td>01b</td> <td>200 ns</td> <td>11b</td> <td>800 ns</td> </tr> </tbody> </table>	Bits	Description	Bits	Description	00b	100 ns	10b	400 ns	01b	200 ns	11b	800 ns
Bits	Description	Bits	Description										
00b	100 ns	10b	400 ns										
01b	200 ns	11b	800 ns										
9	<p><b>FastSlamTimeDown.</b> Read-write. Cold reset: 0. BIOS: 1. Specifies the time the processor waits for downward voltage transitions to complete. This field only effects transitions from <a href="#">D18F4x16C</a>[CstateBoost] or lower performance P-states. 0=<a href="#">D18F3xD8</a>[VSRampSlamTime] . 1=10 us.</p>												
8:7	<p><b>PllRegTime: Pll regulator time.</b> Read-write. Cold reset: 10b. The VDDA regulator may be powered down when the processor transitions to PC6 or CC6. See <a href="#">PC6PwrDwnRegEn</a> and <a href="#">CC6PwrDwnRegEn</a>. If <a href="#">CC6PwrDwnRegEn</a>=1, the VDDA regulator is powered down during CC6. If <a href="#">PC6PwrDwnRegEn</a>=1, the VDDA regulator is powered down during PC6. If the VDDA regulator is powered down during CC6 and the core transitions from CC6 to PC6, the regulator remains powered down during PC6 regardless of the <a href="#">PC6PwrDwnRegEn</a> setting. This field specifies the time required for the VDDA regulator to power back up and initialize the core PLL logic that is powered by the VDDA regulator.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Reserved.</td> <td>10b</td> <td>1.5 us</td> </tr> <tr> <td>01b</td> <td>Reserved.</td> <td>11b</td> <td>2.0 us</td> </tr> </tbody> </table>	Bits	Description	Bits	Description	00b	Reserved.	10b	1.5 us	01b	Reserved.	11b	2.0 us
Bits	Description	Bits	Description										
00b	Reserved.	10b	1.5 us										
01b	Reserved.	11b	2.0 us										
6:0	<p><b>PC6Vid[6:0]: package C6 vid.</b> Read-write. Cold reset: Product-specific. <a href="#">PC6Vid</a>[7:0] = {<a href="#">PC6Vid</a>[7], <a href="#">PC6Vid</a>[6:0]}. <a href="#">PC6Vid</a>[7:0] specifies the VID driven in the PC6 state. See <a href="#">2.5.3.2.3.4 [Package C6 (PC6) State]</a> and <a href="#">2.5.1.3.2 [Low Power Voltages]</a>.</p>												

#### D18F5x12C Clock Power/Timing Control 4

See the *AMD Serial VID Interface 2.0 (SVI2) Specification*.

Bits	Description
31	<p><b>Svi2CmdBusy.</b> Read-only, updated-by-hardware. Cold reset: 0. 1=SVI2 command in progress. This bit is set by hardware when any SVI2 command is sent to the voltage regulator. Software must wait for this bit to clear to 0 before writing any of the following fields: <a href="#">D18F5x12C</a>[CorePsi1En, CoreLoadLineTrim, CoreOffsetTrim], <a href="#">D18F5x188</a>[NbPsi1, NbLoadLineTrim, NbOffsetTrim], <a href="#">D18F5x18C</a>[CoreTfn, NbTfn]. This bit is cleared by hardware when the SVI2 command is complete. On a voltage change, this bit is cleared when the voltage transition is completed. See <a href="#">2.5.1.4.1 [Hardware-Initiated Voltage Transitions]</a>. On a telemetry or PSIX_L change, this bit is cleared as soon as the SVI2 command is sent to the voltage regulator. See <a href="#">2.5.1.1.1 [SVI2 Features]</a> and <a href="#">2.5.1.3.1 [PSIX_L Bit]</a>.</p>
30	<p><b>WaitVidCompDis: wait VID completion disable.</b> IF (<a href="#">D18F2x1B4</a>[SmuCfgLock]) THEN Read-only; updated-by-hardware. ELSE Read-write. ENDIF. Cold reset: 0. 0=Hardware waits for the VOTF complete indicator from the voltage regulator before clearing Svi2CmdBusy or making additional voltage change requests. 1=Hardware clears Svi2CmdBusy 500us after changes to <a href="#">CoreLoadLineTrim</a>, <a href="#">CoreOffsetTrim</a>, or <a href="#">D18F5x188</a>[NbLoadLineTrim, NbOffsetTrim] are made; hardware clears Svi2CmdBusy and additional voltage changes are allowed after the time specified by <a href="#">D18F3xD8</a>[VSRampSlamTime] . See <a href="#">2.5.1.4 [Voltage Transitions]</a>.</p>
29:6	RAZ.

5	<b>CorePsi1En: Core PSII_L enable.</b> IF (D18F2x1B4[SmuCfgLock]) THEN Read-only; updated-by-hardware. ELSE Read-write. ENDIF Cold reset: 0. BIOS: 1. 0=PSII_L for VDD is deasserted. 1=PSII_L for VDD is asserted when all cores are in CC6. See 2.5.3.2.3.4 [Package C6 (PC6) State], 2.5.1.3.1 [PSIx_L Bit], and Svi2CmdBusy.																				
4:2	<p><b>CoreLoadLineTrim: Core load line trim.</b> IF (D18F2x1B4[SmuCfgLock]) THEN Read-only; updated-by-hardware. ELSE Read-write. ENDIF. Cold reset: 011b. CoreLoadLineTrim and NbLoadLineTrim specify a percentage change relative to the initial load line slope for VDD and VDDNB, respectively. See Svi2CmdBusy.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>Load line disabled</td> <td>100b</td> <td>+20%</td> </tr> <tr> <td>001b</td> <td>-40%</td> <td>101b</td> <td>+40%</td> </tr> <tr> <td>010b</td> <td>-20%</td> <td>110b</td> <td>+60%</td> </tr> <tr> <td>011b</td> <td>0%</td> <td>111b</td> <td>+80%</td> </tr> </tbody> </table>	Bits	Description	Bits	Description	000b	Load line disabled	100b	+20%	001b	-40%	101b	+40%	010b	-20%	110b	+60%	011b	0%	111b	+80%
Bits	Description	Bits	Description																		
000b	Load line disabled	100b	+20%																		
001b	-40%	101b	+40%																		
010b	-20%	110b	+60%																		
011b	0%	111b	+80%																		
1:0	<p><b>CoreOffsetTrim: Core offset trim.</b> IF (D18F2x1B4[SmuCfgLock]) THEN Read-only; updated-by-hardware. ELSE Read-write. ENDIF. Cold reset: 10b. CoreOffsetTrim and NbOffsetTrim specify a voltage offset relative to the initial load line offset for VDD and VDDNB, respectively. See Svi2CmdBusy.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Load line offset disabled</td> <td>10b</td> <td>0mV</td> </tr> <tr> <td>01b</td> <td>-25mV</td> <td>11b</td> <td>+25mV</td> </tr> </tbody> </table>	Bits	Description	Bits	Description	00b	Load line offset disabled	10b	0mV	01b	-25mV	11b	+25mV								
Bits	Description	Bits	Description																		
00b	Load line offset disabled	10b	0mV																		
01b	-25mV	11b	+25mV																		

**D18F5x16[C:0] Northbridge P-state [3:0]**

Each of these registers specify the frequency and voltage associated with each of the NB P-states.

**Table 177: Register Mapping for D18F5x16[C:0]**

Register	Function
D18F5x160	NB P-state 0
D18F5x164	NB P-state 1
D18F5x168	NB P-state 2
D18F5x16C	NB P-state 3

The NbVid field is allowed to be different between processors in a multi-processor system. All other fields are required to be programmed to the same value for all processors in the coherent fabric. See 2.5.4.1 [NB P-states] for more information about these registers.

**Table 178: NB P-state Definitions**

Term	Definition
<b>NBCOF</b>	NB current operating frequency in MHz. $NBCOF = 100 * (D18F5x16[C:0][NbFid] + 4h) / (2^D18F5x16[C:0][NbDid])$ .
<b>NBCOF[0]</b>	NB current operating frequency in MHz for NB P-state 0. $NBCOF[0] = (100 * (D18F5x160[NbFid] + 4h) / (2^D18F5x160[NbDid]))$ .

**Table 178: NB P-state Definitions**

Term	Definition
<b>NBCOF[1]</b>	NB current operating frequency in MHz for NB P-state 1. $\text{NBCOF}[1] = (100 * (\text{D18F5x164}[\text{NbFid}] + 4\text{h}) / (2^{\text{D18F5x164}[\text{NbDid}]})$
<b>NBCOF[2]</b>	NB current operating frequency in MHz for NB P-state 2. $\text{NBCOF}[2] = (100 * (\text{D18F5x168}[\text{NbFid}] + 4\text{h}) / (2^{\text{D18F5x168}[\text{NbDid}]})$
<b>NBCOF[3]</b>	NB current operating frequency in MHz for NB P-state 3. $\text{NBCOF}[3] = (100 * (\text{D18F5x16C}[\text{NbFid}] + 4\text{h}) / (2^{\text{D18F5x16C}[\text{NbDid}]})$

Bits	Description										
31:24	<b>NbIddValue: northbridge current value.</b> Read-write. Cold reset: Product-specific. See NbIddDiv.										
23:22	<b>NbIddDiv: northbridge current divisor.</b> Read-write. Cold reset: Product-specific. After reset, NbIddDiv and NbIddValue combine to specify the expected maximum current drawn on the VDDNB power plane at a given VDDNB voltage. These values are intended to be used by <a href="#">2.5.1.3.1.1 [BIOS Requirements for PSIO_L]</a> . These values are not intended to convey final product power levels and may not match the power levels specified in the Power and Thermal Datasheet. These fields may be subsequently altered by software; they do not affect the hardware behavior. <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>IddValue / 1 A, Range: 0 to 255 A.</td> </tr> <tr> <td>01b</td> <td>IddValue / 10 A, Range: 0 to 25.5 A.</td> </tr> <tr> <td>10b</td> <td>IddValue / 100 A, Range: 0 to 2.55 A.</td> </tr> <tr> <td>11b</td> <td>Reserved.</td> </tr> </tbody> </table>	Bits	Description	00b	IddValue / 1 A, Range: 0 to 255 A.	01b	IddValue / 10 A, Range: 0 to 25.5 A.	10b	IddValue / 100 A, Range: 0 to 2.55 A.	11b	Reserved.
Bits	Description										
00b	IddValue / 1 A, Range: 0 to 255 A.										
01b	IddValue / 10 A, Range: 0 to 25.5 A.										
10b	IddValue / 100 A, Range: 0 to 2.55 A.										
11b	Reserved.										
21	<b>NbVid[7].</b> Read-write. Cold reset: Product-specific. See NbVid[6:0].										
20	Reserved.										
19	Reserved.										
18	<b>MemPstate: Memory P-state.</b> Read-write. Cold reset: Product-specific. 1=The Northbridge P-state specified by this register maps to memory P-state 1. 0=The Northbridge P-state specified by this register maps to memory P-state 0. Memory P-states may be globally disabled by programming <a href="#">D18F5x170[MemPstateDis]</a> . See <a href="#">2.5.7.1 [Memory P-states]</a> .										
17	Reserved.										
16:10	<b>NbVid[6:0]: Northbridge VID.</b> Read-write. Cold reset: Product-specific. NbVid[7:0] = {NbVid[7], NbVid[6:0]}. NbVid[7:0] specifies the Northbridge voltage.										
9:8	Reserved.										
7	<b>NbDid: Northbridge divisor ID.</b> Read-write. Cold reset: Product-specific. Specifies the Northbridge frequency divisor; see NbFid.										
6:1	<b>NbFid[5]: Northbridge frequency ID.</b> Read-write. Cold reset: Product-specific. Specifies the Northbridge frequency multiplier. The NB COF is a function of NbFid and NbDid, and defined by <b>NBCOF</b> . NbFid and NbDid are not changed on a write if the value written results in a frequency greater than <a href="#">MSRC001_0071[MaxNbCof]</a> . See <a href="#">2.5.3.1.5 [Core P-state Transition Behavior]</a> .										

0	<b>NbPstateEn: Northbridge P-state enable.</b> Read-write. Cold reset: Product-specific. 1=The Northbridge P-state specified by this register is valid. 0=The Northbridge P-state specified by this register is not valid. This bit must be set to 1 in order for the Northbridge P-state specified by this register to be programmed in <a href="#">D18F5x170[NbPstateHi, NbPstateLo]</a> .
---	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**D18F5x170 Northbridge P-state Control**

See also [2.5.4.1 \[NB P-states\]](#).

Bits	Description																				
31	<b>MemPstateDis: memory P-state disable.</b> IF ( <a href="#">D18F3xE8[MemPstateCap]</a> && <a href="#">D18F2x1B4[SmuCfgLock]</a> ==0) THEN Read-write, updated-by-hardware; Updated-by-SMU. ELSE Read-only, updated-by-hardware; Updated-by-SMU. Reset: Product-specific. 1=Memory P-state transitions are disabled. The current P-state is not changed by programming this bit. The memory P-state will be forced to M0 on the next NB P-state transition. On processors where memory P-states are enabled, programming this bit may result in a violation of bandwidth requirements stated in <a href="#">2.5.3.1.5</a> . Software must ensure that NB P-states which violate those requirements are forced disabled. 0=Memory P-state transitions are enabled if <a href="#">D18F2x90_dct[0][DisDIIShutdownSR]</a> =0.																				
30	<b>NbPstateFidVidSbcEn.</b> IF ( <a href="#">D18F5x174[NbPstateDis]</a>    <a href="#">D18F2x1B4[SmuCfgLock]</a> ) THEN Read-only. ELSE Read: Write-1-only. ENDIF. Reset: 0. BIOS: 1. NB P-state transitions are blocked until this field is set to a 1.																				
29:27	<b>NbPstateHiRes: NB P-state high residency timer.</b> If <a href="#">D18F2x1B4[SmuCfgLock]</a> THEN Read-only; updated-by-hardware. ELSE Read-write. ENDIF Reset: 0. Specifies the minimum time the processor must spend in the high NB P-state before transitions to the low NB P-state are allowed. See <a href="#">2.5.4.1 [NB P-states]</a> . <table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center; border-bottom: 1px solid black;">Bits</th> <th style="text-align: center; border-bottom: 1px solid black;">Description</th> <th style="text-align: center; border-bottom: 1px solid black;">Bits</th> <th style="text-align: center; border-bottom: 1px solid black;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">000b</td> <td style="text-align: center;">0us</td> <td style="text-align: center;">100b</td> <td style="text-align: center;">1ms</td> </tr> <tr> <td style="text-align: center;">001b</td> <td style="text-align: center;">10us</td> <td style="text-align: center;">101b</td> <td style="text-align: center;">5ms</td> </tr> <tr> <td style="text-align: center;">010b</td> <td style="text-align: center;">100us</td> <td style="text-align: center;">110b</td> <td style="text-align: center;">10ms</td> </tr> <tr> <td style="text-align: center;">011b</td> <td style="text-align: center;">500us</td> <td style="text-align: center;">111b</td> <td style="text-align: center;">50ms</td> </tr> </tbody> </table>	Bits	Description	Bits	Description	000b	0us	100b	1ms	001b	10us	101b	5ms	010b	100us	110b	10ms	011b	500us	111b	50ms
Bits	Description	Bits	Description																		
000b	0us	100b	1ms																		
001b	10us	101b	5ms																		
010b	100us	110b	10ms																		
011b	500us	111b	50ms																		
26:24	<b>NbPstateLoRes: NB P-state low residency timer.</b> If <a href="#">D18F2x1B4[SmuCfgLock]</a> THEN Read-only; updated-by-hardware. ELSE Read-write. ENDIF Reset: 0. Specifies the minimum time the processor must spend in the low NB P-state before transitions to the high NB P-state are allowed. See <a href="#">2.5.4.1 [NB P-states]</a> . See: <a href="#">NbPstateHiRes</a> .																				
23	<b>NbPstateGnbSlowDis.</b> If <a href="#">D18F2x1B4[SmuCfgLock]</a> THEN Read-only; updated-by-hardware. ELSE Read-write. ENDIF Reset: 0. Specifies whether NBP-state transitions take the GnbSlow signal into account. 0=Take GnbSlow into account. 1=Ignore GnbSlow. See <a href="#">2.5.4.1 [NB P-states]</a> .																				
22:15	Reserved.																				
14	<b>SwNbPstateLoDis: software NB P-state low disable.</b> IF ( <a href="#">D18F5x174[NbPstateDis]</a>   <a href="#">D18F2x1B4[SmuCfgLock]</a> ) THEN Read-only. ELSE Read-write. ENDIF. Reset: 0. 1=Transition to <a href="#">NbPstateHi</a> and disable transitions to <a href="#">NbPstateLo</a> .																				
13	<b>NbPstateDisOnP0: NB P-state disable on P0.</b> IF ( <a href="#">D18F5x174[NbPstateDis]</a>   <a href="#">D18F2x1B4[SmuCfgLock]</a> ) THEN Read-only. ELSE Read-write. ENDIF. Reset: 0. 1=Transition to <a href="#">NbPstateHi</a> and disable transitions to <a href="#">NbPstateLo</a> if any compute unit is in P0 or a boosted P-state. This field uses software P-state numbering. See <a href="#">2.5.3.1.1.1 [Software P-state Numbering]</a> .																				



12:9	<b>NbPstateThreshold: NB P-state threshold.</b> If <a href="#">D18F2x1B4[SmuCfgLock]</a> THEN Read-only; updated-by-hardware. ELSE Read-write. ENDIF Reset: Product-specific. BIOS: <a href="#">COUNT(D18F5x80[Enabled])</a> . Specifies the minimum number of compute units that must be in a P-state with <a href="#">MSRC001_00[6B:64][NbPstate]=1</a> before transitions to lower performance NB P-states are allowed. See <a href="#">NbPstateLo</a> and <a href="#">NbPstateHi</a> .
8	Reserved.
7:6	<b>NbPstateHi: NB P-state high.</b> IF ( <a href="#">D18F2x1B4[SmuCfgLock]</a> ) THEN Read-only; updated-by-hardware. ELSE Read-write. ENDIF. Cold reset: Product-specific. If NB P-states are enabled, this field specifies the NB P-state that is used when the number of compute units in a P-state with <a href="#">MSRC001_00[6B:64][NbPstate]=1</a> is less than <a href="#">NbPstateThreshold</a> . This field must be programmed to the same value for all processors in the coherent fabric. This field is not changed on a write if the value written is greater than the <a href="#">NbPstateMaxVal</a> value written or greater than the current <a href="#">NbPstateLo</a> value. See also <a href="#">NbPstateDisOnP0</a> , <a href="#">SwNbPstateLoDis</a> , <a href="#">NbPstateLo</a> , <a href="#">D18F5x174[NbPstateDis]</a> , and <a href="#">D18F5x16[C:0][NbPstateEn]</a> .
5	Reserved.
4:3	<b>NbPstateLo: NB P-state low.</b> IF ( <a href="#">D18F2x1B4[SmuCfgLock]</a> ) THEN Read-only; updated-by-hardware. ELSE Read-write. ENDIF. Cold reset: Product-specific. If NB P-states are enabled, this field specifies the NB P-state that is used when the number of compute units in a P-state with <a href="#">MSRC001_00[6B:64][NbPstate]=1</a> is greater than or equal to <a href="#">NbPstateThreshold</a> . <a href="#">NbPstateLo</a> must be greater than or equal to <a href="#">NbPstateHi</a> . This field must be programmed to the same value for all processors in the coherent fabric. This field is not changed on a write if the value written is greater than the <a href="#">NbPstateMaxVal</a> value written or less than the current <a href="#">NbPstateHi</a> value. See also <a href="#">NbPstateDisOnP0</a> , <a href="#">SwNbPstateLoDis</a> , <a href="#">D18F5x174[NbPstateDis]</a> , and <a href="#">D18F5x16[C:0][NbPstateEn]</a> .
2	Reserved.
1:0	<b>NbPstateMaxVal: NB P-state maximum value.</b> IF ( <a href="#">D18F2x1B4[SmuCfgLock]</a> ) THEN Read-only; updated-by-hardware. ELSE Read-write. ENDIF. Cold reset: specified by the reset state of <a href="#">D18F5x16[C:0][NbPstateEn]</a> ; the cold reset value is the highest NB P-state number corresponding to the register in which <a href="#">NbPstateEn</a> is set (e.g., if <a href="#">D18F5x160</a> and <a href="#">D18F5x164</a> have this bit set and the others do not, then <a href="#">NbPstateMaxVal</a> =1; if <a href="#">NbPstateEn</a> is only set in <a href="#">D18F5x160</a> , then <a href="#">NbPstateMaxVal</a> =0). This specifies the highest NB P-state value (lowest performance state) supported by the hardware.

### D18F5x174 Northbridge P-state Status

Bits	Description
31:25	Reserved.
24	<b>CurMemPstate: current memory P-state.</b> Read-only; updated-by-hardware. Reset: 0. Specifies the current memory P-state. 1=Memory P-state 1. 0=Memory P-state 0. See <a href="#">2.5.7.1 [Memory P-states]</a> .
23	<b>CurNbVid[7]: current northbridge voltage ID[7].</b> <a href="#">MSRC001_0071[CurNbVid[7]]</a> is an alias of <a href="#">D18F5x174[CurNbVid[7]]</a> . . VDDNB voltage.
22	<b>CurNbPstateLo.</b> Read-only; updated-by-hardware. Reset: 0. 1=Current NB Pstate maps to <a href="#">D18F5x170[NbPstateLo]</a> . 0=Current NB Pstate maps to <a href="#">D18F5x170[NbPstateHi]</a> .
21	Reserved.

20:19	<b>CurNbPstate: current northbridge P-state.</b> Read-only; updated-by-hardware. Reset: 0. Provides the NB P-state that corresponds to the current frequency component of the NB. The value of this field is updated when the COF transitions to a new value associated with an NB P-state. <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>NB P0</td> </tr> <tr> <td>01b</td> <td>NB P1</td> </tr> <tr> <td>1xb</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Description	00b	NB P0	01b	NB P1	1xb	Reserved
Bits	Description								
00b	NB P0								
01b	NB P1								
1xb	Reserved								
18:12	<b>CurNbVid[6:0]: current northbridge voltage ID.</b> <a href="#">MSRC001_0071</a> [CurNbVid[6:0]] is an alias of <a href="#">D18F5x174</a> [CurNbVid[6:0]]. VDDNB voltage.								
11	Reserved.								
10	Reserved.								
9	<b>CurNbDid: current northbridge divisor ID.</b> Read-only, updated-by-hardware. Reset: 0.								
8:3	<b>CurNbFid[5:0]: current northbridge frequency ID.</b> Read-only, updated-by-hardware. Reset: 0.								
2:1	<b>StartupNbPstate: startup northbridge P-state number.</b> Read-only. Cold reset: Product-specific. Specifies the cold reset VID, FID and DID for the Northbridge based on the NB P-state number selected.								
0	<b>NbPstateDis: northbridge P-state disable.</b> Read-only. Value: Product-specific. <a href="#">MSRC001_0071</a> [NbPstateDis] is an alias of <a href="#">D18F5x174</a> [NbPstateDis].								

### D18F5x178 Northbridge Fusion Configuration

Bits	Description
31:20	Reserved.
19	<b>SwGfxDis.</b> Read-write. Reset: 1. BIOS: IF ( <a href="#">GpuEnabled</a> ) THEN 0 ELSE 1 ENDIF. 1=Hardware handshakes for NB P-state transitions and DRAM self-refresh entry are ignored. See <a href="#">2.5.4.1.1 [NB P-state Transitions]</a> . See <a href="#">2.5.7.2 [DRAM Self-Refresh]</a> .
18	<b>CstateFusionHsDis: C-state fusion handshake disable.</b> Read-write. Reset: 0. BIOS:1. 1=Ignore the FCH handshake response for PC6 transitions. 0=Use the FCH handshake response for PC6 entry. See <a href="#">2.5.3.2.4.1 [FCH Messaging]</a> .
17	<b>Dis2ndGnbAllowPsWait.</b> Read-write. Reset: 0. BIOS: 1. 1=Do not do a second check of AllowNbTrans after quiescing the cores when transitioning NB P-states. See <a href="#">2.5.4.1.1 [NB P-state Transitions]</a> .
16	<b>ProcHotToGnbEn.</b> Read-write. Reset: 0. BIOS: 1. 1=The GPU is placed into a low-power state when PROCHOT_L is asserted. Note: the GPU power-state transitions associated with PROCHOT_L nominally occur every 1 millisecond; PROCHOT_L assertions and deassertions for less than this period may not result in GPU state changes.
15:12	Reserved.
11	<b>AllowSelfRefrS3Dis: allow self-refresh S3 disable.</b> Read-write. Reset: 0. BIOS: 1. 1=The NB does not wait for handshake before placing DRAM into self-refresh (see <a href="#">2.5.7.2 [DRAM Self-Refresh]</a> ) on S3 entry (see <a href="#">2.5.8.1.1 [ACPI Suspend to RAM State (S3)]</a> ). 0=The NB waits for handshake before placing DRAM into self-refresh on S3 entry.

10	<b>InbWakeS3Dis: InbWake S3 disable.</b> Read-write. Reset: 0. BIOS: 1. 1= The NB does not wait for handshake before placing DRAM into self-refresh (see 2.5.7.2 [DRAM Self-Refresh]) on S3 entry (see 2.5.8.1.1 [ACPI Suspend to RAM State (S3)]). 0=The NB waits for handshake before placing DRAM into self-refresh on S3 entry.
9:4	Reserved.
3	<b>CstateThreeWayHsEn: C-state three way handshake disable.</b> Read-write. Reset: 0. 1=Enable the three way handshake with the FCH when entering a C-state. 0=Only a two way handshake with FCH is used. There is no message about the resulting package state sent to FCH. See 2.5.3.2.4.1 [FCH Messaging].
2	<b>CstateFusionDis: C-state fusion disable.</b> Read-write. Reset: 0. 1=All HALT or C-state requests are forwarded to the FCH. 0=HALT and C-state requests are forwarded to the FCH when each core has made a request. See 2.5.3.2.4.1 [FCH Messaging].
1:0	Reserved.

### D18F5x17C Miscellaneous Voltages

Bits	Description
31	<b>NbPsi0VidEn: Northbridge PSIO_L VID enable.</b> Read-write. Reset: 0. This bit specifies how PSIO_L is controlled for VDDNB. See D18F3xA0[PsiVidEn] and 2.5.1.3.1 [PSIx_L Bit].
30:23	<b>NbPsi0Vid[7:0]: Northbridge PSIO_L VID threshold.</b> Read-write. Reset: 0. When enabled by NbPsi0VidEn, NbPsi0Vid specifies the threshold value of the VID code generated by the Northbridge, which in turn determines the state of PSIO_L. See D18F3xA0[PsiVid[6:0]] and 2.5.1.3.1 [PSIx_L Bit].
22:18	Reserved.
17:10	<b>MinVid: minimum voltage.</b> 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].
9:8	Reserved.
7:0	<b>MaxVid: maximum voltage.</b> 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].

### D18F5x188 Clock Power/Timing Control 5

See the *AMD Serial VID Interface 2.0 (SVI2) Specification*.

Bits	Description
31:6	RAZ.
5	<b>NbPsi1: Northbridge PSII_L.</b> IF D18F2x1B4[SmuCfgLock] THEN Read-only; updated-by-hardware. ELSE Read-write. ENDIF Cold reset: 0. Specifies how PSII_L is controlled for VDDNB. 1=PSII_L is low. 0=PSII_L is high. See 2.5.1.3.1 [PSIx_L Bit].
4:2	<b>NbLoadLineTrim: Northbridge load line trim.</b> IF D18F2x1B4[SmuCfgLock] THEN Read-only; updated-by-hardware. ELSE Read-write. ENDIF Cold reset: 011b. See D18F5x12C[CoreLoadLine-Trim].

1:0	<b>NbOffsetTrim: Northbridge offset trim.</b> IF <a href="#">D18F2x1B4</a> [SmuCfgLock] THEN Read-only; updated-by-hardware. ELSE Read-write. ENDIF. Cold reset: 10b. See <a href="#">D18F5x12C</a> [CoreOffsetTrim].
-----	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### D18F5x18C Clock Power/Timing Control 6

See the *AMD Serial VID Interface 2.0 (SVI2) Specification*.

Bits	Description															
31:2	RAZ.															
1	<b>CoreTfn: Core telemetry functionality.</b> If <a href="#">D18F2x1B4</a> [SmuCfgLock] THEN Read-only; updated-by-hardware. ELSE Read-write. ENDIF Cold reset: 0. See NbTfn.															
0	<b>NbTfn: Northbridge telemetry functionality.</b> If <a href="#">D18F2x1B4</a> [SmuCfgLock] THEN Read-only; updated-by-hardware. ELSE Read-write. ENDIF. Cold reset: 0. See <a href="#">D18F5x12C</a> [Svi2CmdBusy]. CoreTfn and NbTfn specify the telemetry mode as follows:															
	<table border="1"> <thead> <tr> <th>CoreTfn</th> <th>NbTfn</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Telemetry enabled in voltage-only mode.</td> </tr> <tr> <td>0</td> <td>1</td> <td>Telemetry enabled in voltage and current mode.</td> </tr> <tr> <td>1</td> <td>0</td> <td>Telemetry disabled.</td> </tr> <tr> <td>1</td> <td>1</td> <td>Reserved.</td> </tr> </tbody> </table>	CoreTfn	NbTfn	Description	0	0	Telemetry enabled in voltage-only mode.	0	1	Telemetry enabled in voltage and current mode.	1	0	Telemetry disabled.	1	1	Reserved.
CoreTfn	NbTfn	Description														
0	0	Telemetry enabled in voltage-only mode.														
0	1	Telemetry enabled in voltage and current mode.														
1	0	Telemetry disabled.														
1	1	Reserved.														

### D18F5x194 Name String Address Port

[D18F5x194](#) and [D18F5x198](#) provide BIOS with a read-only name string that may be copied to [MSRC001\\_00](#)[35:30] at warm reset. Each of [D18F5x198\\_x](#)[B:0] is read as follows:

- Write [D18F5x194](#)[Index].
- Read [D18F5x198](#).

Bits	Description						
31:4	Reserved.						
3:0	<b>Index: name string register index.</b> Read-write. Reset: 0.						
	<table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Bh-0h</td> <td>Name String Registers. See <a href="#">D18F5x198_x</a>[B:0].</td> </tr> <tr> <td>Fh-Ch</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Description	Bh-0h	Name String Registers. See <a href="#">D18F5x198_x</a> [B:0].	Fh-Ch	Reserved
Bits	Description						
Bh-0h	Name String Registers. See <a href="#">D18F5x198_x</a> [B:0].						
Fh-Ch	Reserved						

### D18F5x198 Name String Data Port

See [D18F5x194](#) for register access information. Address: [D18F5x194](#)[Index].

Bits	Description
31:0	<b>Data.</b>

### D18F5x198\_x[B:0] Name String Data

Bits	Description
31:24	<b>NameStringByte3: name string ASCII character 3.</b> Read-only. Value: Product-specific.
23:16	<b>NameStringByte2: name string ASCII character 2.</b> Read-only. Value: Product-specific.

15:8	<b>NameStringByte1: name string ASCII character 1.</b> Read-only. Value: Product-specific.
7:0	<b>NameStringByte0: name string ASCII character 0.</b> Read-only. Value: Product-specific.

### D18F5x240 ECC Exclusion Base Address Low

- Transaction addresses are within the defined range if:  
{EccExclBaseAddr[47:6], 00\_0000b} <= address[47:0] <= {EccExclLimitAddr[47:6], 00\_0000b}.
- BIOS must quiesce all other forms of DRAM traffic when configuring this range. See [MSRC001\\_001F](#)[DisDramScrub].
- When initializing the base/limit pair, the BIOS must write the limit register before the EccExclEn bit is set. BIOS should clear EccExclEn before changing the address range.
- BIOS should take care to re-initialize memory with valid ECC when resizing this region.

Bits	Description
31:6	<b>EccExclBaseAddr[31:6]: ECC exclusion base address register bits[31:6].</b> Read-write. Reset: 0. EccExclBaseAddr[47:6]={ <a href="#">D18F5x244</a> [EccExclBaseAddr[47:32]], EccExclBaseAddr[31:6]}. The ECC Exclusion Base/Limit Address registers setup a contiguous range in DRAM where ECC check and error reporting is disabled. BIOS configures the ECC exclusion range code to cover the frame buffer region in ECC UMA systems with internal GPUs. The GPU is configured as MC_SHARED:MC_VM_STEERING [DEFAULT_STEERING]=1 (system traffic to onion).
5:1	Reserved.
0	<b>EccExclEn.</b> Read-write. Reset: 0. 1=Enable ECC Exclusion Range. See <a href="#">D18F5x240</a> [EccExclBaseAddr].

### D18F5x244 ECC Exclusion Base Address High

Bits	Description
31:16	Reserved.
15:0	<b>EccExclBaseAddr[47:32]: ECC exclusion base address register bits[47:32].</b> Read-write. Reset: 0. See <a href="#">D18F5x240</a> [EccExclBaseAddr].

### D18F5x248 ECC Exclusion Limit Address Low

Bits	Description
31:6	<b>EccExclLimitAddr[31:6]: ECC exclusion limit address register bits[31:6].</b> Read-write. Reset: 0. EccExclLimitAddr[47:6]={ <a href="#">D18F5x24C</a> [EccExclLimitAddr[47:32]], EccExclLimitAddr[31:6]}. See <a href="#">D18F5x240</a> [EccExclBaseAddr].
5:0	Reserved.

### D18F5x24C ECC Exclusion Limit Address High

Bits	Description

31:16	Reserved.
15:0	<b>EccExclLimitAddr[47:32]: ECC exclusion limit address register bits[47:32].</b> Read-write. Reset: 0. See <a href="#">D18F5x240[EccExclBaseAddr]</a> .

### **D18F5x260 Clock Power/Timing Control 8**

---

Bits	Description
31:4	Reserved.
3:0	Reserved.

### 3.15 Northbridge IOAPIC Registers

The Northbridge IOAPIC is accessed through the Northbridge IOAPIC base address specified by [D0F0xFC\\_x01 \[IOAPIC Base Address Lower\]](#) and [D0F0xFC\\_x02 \[IOAPIC Base Address Upper\]](#).

#### NBIOAPICx00 IO Register Select

Bits	Description
31:8	Reserved.
7:0	<b>IndirectAddressOffset</b> . Read-write. Reset: 0. Specifies the indexed register accessed via <a href="#">NBIOAPICx10 [IO Window]</a> .

#### NBIOAPICx10 IO Window

Bits	Description
31:0	<b>IoapicData</b> .

#### NBIOAPICx10\_x00 IOAPIC ID

This register is not used in IOxAPIC PCI bus delivery mode.

Bits	Description
31:28	<b>ExtendID: extended IOAPIC device ID</b> . IF ( <a href="#">D0F0xFC_x00</a> [IoapicIdExtEn]==0) THEN Read-only. ELSE Read-write. ENDIF. Reset: 0.
27:24	<b>ID: IOAPIC device ID</b> . Read-write. Reset: 0.
23:0	Reserved.

#### NBIOAPICx10\_x01 IOAPIC Version

Bits	Description
31:24	Reserved.
23:16	<b>MaxRedirectionEntries</b> . Value: 1Fh. Indicates 32 entries [31:0].
15	<b>PRQ</b> . Value: 1. IRQ pin assertion supported
14:8	Reserved.
7:0	<b>Version</b> . Value: 21h. PCI 2.2 compliant

#### NBIOAPICx10\_x02 IOAPIC Arbitration

Bits	Description
31:28	Reserved.
27:24	<b>ArbitrationID</b> . Read-only. Reset: 0.
23:0	Reserved.

**NBIOAPICx10\_x[4E:10:step2] Redirection Table Entry [31:0]**

Bits	Description																				
63:56	<b>DestinationID.</b> Read-write. Reset: 0. Bits [19:12] of the address field of the interrupt message.																				
55:32	Reserved.																				
31:17	Reserved.																				
16	<b>Mask.</b> Read-write. Reset: 1. 1=Mask the interrupt injection at the input of this device. 0=Unmask.																				
15	<b>TriggerMode.</b> Read-write. Reset: 0. 0=Edge. 1=Level																				
14	<b>RemoteIRR.</b> Read-only. Reset: 0. Used for level triggered interrupts only. It is cleared by EOI special cycle transaction or write to EOI register. 1=Interrupt message is delivered.																				
13	<b>InterruptPinPolarity.</b> Read-write. Reset: 0. 0=High. 1=Low.																				
12	<b>DeliveryStatus.</b> Read-only. Reset: 0. 0=Idle. 1=Send Pending.																				
11	<b>DestinationMode.</b> Read-write. Reset: 0. 0=Physical. 1=Logical																				
10:8	<b>DeliveryMode.</b> Read-write. Reset: 0. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Bits</th> <th>Definition</th> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <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/PMI</td> <td>110b</td> <td>Reserved</td> </tr> <tr> <td>011b</td> <td>Reserved</td> <td>111b</td> <td>ExtINT</td> </tr> </tbody> </table>	Bits	Definition	Bits	Definition	000b	Fixed	100b	NMI	001b	Lowest Priority	101b	INIT	010b	SMI/PMI	110b	Reserved	011b	Reserved	111b	ExtINT
Bits	Definition	Bits	Definition																		
000b	Fixed	100b	NMI																		
001b	Lowest Priority	101b	INIT																		
010b	SMI/PMI	110b	Reserved																		
011b	Reserved	111b	ExtINT																		
7:0	<b>Vector.</b> Read-write. Reset: 0. Interrupt vector associated with this interrupt input																				

**NBIOAPICx20 IRQ Pin Assertion**

Bits	Description
31:8	Reserved.
7:0	<b>InputIrq.</b> Read-write. Reset: 0. IRQ number for the requested interrupt. A write to this register will trigger an interrupt associated with the redirection table entry referenced by the IRQ number. Currently the redirection table has 24 entries. Writes with IRQ number greater than 17h have no effect.

**NBIOAPICx40 EOI**

Bits	Description
31:8	Reserved.
7:0	<b>Vector.</b> Write-only. Reset: 0. Interrupt vector. A write to this register will clear the remote IRR bit in the redirection table entry found matching the interrupt vector. This provides an alternate mechanism other than PCI special cycle for EOI to reach IOxAPIC.



### 3.16 APIC Registers

See 2.4.8.1.2 [APIC Register Space].

MMIO local APIC space is accessible in xAPIC mode.

#### APIC20 APIC ID

Bits	Description
31:24	<b>ApicId: APIC ID.</b> Read-write. Reset: Varies based on core number. <ul style="list-style-type: none"> <li>The initial value of APIC20[ApicId[7:0]] is {0000b, CpuCoreNum[3:0]}.</li> </ul> See 2.4.8.1.3 [ApicId Enumeration Requirements]. See 2.4.4 [Processor Cores and Downcoring].
23:0	Reserved.

#### APIC30 APIC Version

Bits	Description
31	<b>ExtApicSpace: extended APIC register space present.</b> Read-only. Reset: 1. 1=Indicates the presence of extended APIC register space starting at APIC400.
30:25	RAZ.
24	<b>DirectedEoiSupport: directed EOI support.</b> Read-only. Reset: 0. 0=Directed EOI capability not supported.
23:16	<b>MaxLvtEntry.</b> Read-only. Reset: Product-specific. Specifies the number of entries in the local vector table minus one.
15:8	RAZ.
7:0	<b>Version.</b> Read-only. Reset: 10h. Indicates the version number of this APIC implementation.

#### APIC80 Task Priority (TPR)

Bits	Description
31:8	RAZ.
7:0	<b>Priority.</b> Read-write. Reset: 0. This field is assigned by software to set a threshold priority at which the core is interrupted.

#### APIC90 Arbitration Priority (APR)

Bits	Description
31:8	RAZ.
7:0	<b>Priority.</b> Read-only. Reset: 0. 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 (PPR)**

Bits	Description
31:8	RAZ.
7:0	<b>Priority.</b> Reset: 0. Read-only. 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**

This register is written by the software interrupt handler to indicate the servicing of the current interrupt is complete.

Bits	Description
31:0	Unused. Write-only.

**APICC0 Remote Read**

Bits	Description
31:0	<b>RemoteReadData.</b> Read-only. Reset: 0. The data resulting from a valid completion of a remote read inter-processor interrupt.

**APICD0 Logical Destination (LDR)**

Bits	Description
31:24	<b>Destination.</b> Read-write. Reset: 0. This APIC's destination identification. Used to determine which interrupts should be accepted.
23:0	Reserved.

**APICE0 Destination Format**

Only supported in xAPIC mode.

Bits	Description								
31:28	<b>Format.</b> Read-write. Reset: Fh. Controls which format to use when accepting interrupts with a logical destination mode. <table border="0" style="margin-left: 20px;"> <tr> <td style="padding-right: 20px;"><u>Bits</u></td> <td><u>Definition</u></td> </tr> <tr> <td>0h</td> <td>Cluster destinations are used</td> </tr> <tr> <td>Eh-1h</td> <td>Reserved</td> </tr> <tr> <td>Fh</td> <td>Flat destinations are used</td> </tr> </table>	<u>Bits</u>	<u>Definition</u>	0h	Cluster destinations are used	Eh-1h	Reserved	Fh	Flat destinations are used
<u>Bits</u>	<u>Definition</u>								
0h	Cluster destinations are used								
Eh-1h	Reserved								
Fh	Flat destinations are used								
27:0	Reserved. Reset: FFF_FFFFh.								

**APICF0 Spurious-Interrupt Vector (SVR)**

Bits	Description
31:13	RAZ.
12	<b>EoiBroadcastDisable: EOI broadcast disable.</b> Read-only. Reset: 0.
11:10	RAZ.
9	<b>FocusDisable.</b> Read-write. Reset: 0. 1=Disable focus core checking during lowest-priority arbitrated interrupts.
8	<b>APICSWEn: APIC software enable.</b> Read-write. Reset: 0. 0=SMI, NMI, INIT, LINT[1:0], and Startup interrupts may be accepted; pending interrupts in <b>APIC[170:100]</b> and <b>APIC[270:200]</b> are held, but further fixed, lowest-priority, and ExtInt interrupts are not accepted. All LVT entry mask bits are set and cannot be cleared.
7:0	<b>Vector.</b> Read-write. Reset: FFh. 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 <b>D18F0x68[ApicExtSpur]</b> .

**APIC[170:100] In-Service (ISR)**

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. Interrupts are mapped as follows:

Table 179: [Register Mapping](#) for APIC[170:100]

Register	Function
APIC100	Interrupts [31:16]
APIC110	Interrupts [63:32]
APIC120	Interrupts [95:64]
APIC130	Interrupts [127:96]
APIC140	Interrupts [159:128]
APIC150	Interrupts [191:160]
APIC160	Interrupts [223:192]
APIC170	Interrupts [255:224]

Bits	Description
31:0	<b>InServiceBits.</b> Reset: 0. Read-only. These bits are set when the corresponding interrupt is being serviced by the core.

**APIC[1F0:180] Trigger Mode (TMR)**

The trigger mode registers provide a bit per interrupt to indicate the assertion mode of each interrupt. APIC180[15:0] are reserved. Interrupts are mapped as follows:

Table 180: Register Mapping for APIC[1F0:180]

Register	Function
APIC180	Interrupts [31:16]
APIC190	Interrupts [63:32]
APIC1A0	Interrupts [95:64]
APIC1B0	Interrupts [127:96]
APIC1C0	Interrupts [159:128]
APIC1D0	Interrupts [191:160]
APIC1E0	Interrupts [223:192]
APIC1F0	Interrupts [255:224]

Bits	Description
31:0	<b>TriggerModeBits.</b> Reset: 0. Read-only. The corresponding trigger mode bit is updated when an interrupt is accepted. The values are: 0=Edge-triggered interrupt. 1=Level-triggered interrupt.

### APIC[270:200] Interrupt Request (IRR)

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. Interrupts are mapped as follows:

Table 181: Register Mapping for APIC[270:200]

Register	Function
APIC200	Interrupts [31:16]
APIC210	Interrupts [63:32]
APIC220	Interrupts [95:64]
APIC230	Interrupts [127:96]
APIC240	Interrupts [159:128]
APIC250	Interrupts [191:160]
APIC260	Interrupts [223:192]
APIC270	Interrupts [255:224]

Bits	Description
31:0	<b>RequestBits.</b> Read-only. Reset: 0. The corresponding request bit is set when the an interrupt is accepted by the APIC.

### APIC280 Error Status

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	RAZ.

7	<b>IllegalRegAddr: illegal register address.</b> Read-write. Reset: 0. This bit indicates that an access to a nonexistent register location within this APIC was attempted. Can only be set in xAPIC mode.
6	<b>RcvdIllegalVector: received illegal vector.</b> Read-write. Reset: 0. This bit indicates that this APIC has received a message with an illegal vector (00h to 0Fh for fixed and lowest priority interrupts).
5	<b>SentIllegalVector.</b> Read-write. Reset: 0. 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	RAZ.
3	<b>RcvAcceptError: receive accept error.</b> Read-write. Reset: 0. This bit indicates that a message received by this APIC was not accepted by this or any other APIC.
2	<b>SendAcceptError.</b> Read-write. Reset: 0. This bit indicates that a message sent by this APIC was not accepted by any APIC.
1:0	RAZ.

**APIC300 Interrupt Command Low (ICR Low)**

Not all combinations of ICR fields are valid. Only the following combinations are valid:

**Table 182: ICR valid combinations**

Message Type	Trigger Mode	Level	Destination Shorthand
Fixed	Edge	x	x
	Level	Assert	x
Lowest Priority, SMI, NMI, INIT	Edge	x	Destination or all excluding self.
	Level	Assert	Destination or all excluding self
Startup	x	x	Destination or all excluding self

Note: x indicates a don't care.

Bits	Description										
31:20	RAZ.										
19:18	<p><b>DestShrthnd: destination shorthand.</b> Read-write. Reset: 0. Provides a quick way to specify a destination for a message.</p> <table border="0"> <tr> <td><u>Bits</u></td> <td><u>Description</u></td> </tr> <tr> <td>00b</td> <td>No shorthand (Destination field)</td> </tr> <tr> <td>01b</td> <td>Self</td> </tr> <tr> <td>10b</td> <td>All including self</td> </tr> <tr> <td>11b</td> <td>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.)</td> </tr> </table> <p>If all including self or all excluding self is used, then destination mode is ignored and physical is automatically used.</p>	<u>Bits</u>	<u>Description</u>	00b	No shorthand (Destination field)	01b	Self	10b	All including self	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.)
<u>Bits</u>	<u>Description</u>										
00b	No shorthand (Destination field)										
01b	Self										
10b	All including self										
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.)										

17:16	<b>RemoteRdStat: remote read status.</b> Read-only. Reset: 0. <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Read was invalid</td> </tr> <tr> <td>01b</td> <td>Delivery pending</td> </tr> <tr> <td>10b</td> <td>Delivery complete and access was valid</td> </tr> <tr> <td>11b</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Description	00b	Read was invalid	01b	Delivery pending	10b	Delivery complete and access was valid	11b	Reserved								
Bits	Description																		
00b	Read was invalid																		
01b	Delivery pending																		
10b	Delivery complete and access was valid																		
11b	Reserved																		
15	<b>TM: trigger mode.</b> Read-write. Reset: 0. Indicates how this interrupt is triggered. 0=Edge triggered. 1=Level triggered.																		
14	<b>Level.</b> Read-write. Reset: 0. 0=Deasserted. 1=Asserted.																		
13	RAZ.																		
12	<b>DS: interrupt delivery status.</b> Read-only. Reset: 0. In xAPIC mode this bit is set to indicate that the interrupt has not yet been accepted by the destination core(s). 0=Idle. 1=Send pending. Reserved in x2APIC mode. Software may repeatedly write ICRL without polling the DS bit; all requested IPIs will be delivered.																		
11	<b>DM: destination mode.</b> Read-write. Reset: 0. 0=Physical. 1=Logical.																		
10:8	<b>MsgType.</b> Read-write. Reset: 0. The message types are encoded as follows: <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>Fixed</td> </tr> <tr> <td>001b</td> <td>Lowest Priority.</td> </tr> <tr> <td>010b</td> <td>SMI</td> </tr> <tr> <td>011b</td> <td>Remote read.</td> </tr> <tr> <td>100b</td> <td>NMI</td> </tr> <tr> <td>101b</td> <td>INIT</td> </tr> <tr> <td>110b</td> <td>Startup</td> </tr> <tr> <td>111b</td> <td>External interrupt.</td> </tr> </tbody> </table>	Bits	Description	000b	Fixed	001b	Lowest Priority.	010b	SMI	011b	Remote read.	100b	NMI	101b	INIT	110b	Startup	111b	External interrupt.
Bits	Description																		
000b	Fixed																		
001b	Lowest Priority.																		
010b	SMI																		
011b	Remote read.																		
100b	NMI																		
101b	INIT																		
110b	Startup																		
111b	External interrupt.																		
7:0	<b>Vector.</b> Read-write. Reset: 0. The vector that is sent for this interrupt source.																		

### APIC310 Interrupt Command High (ICR High)

Bits	Description
31:24	<b>DestinationField.</b> Read-write. Reset: 0. The destination encoding used when APIC300[DestShrthnd] is 00b.
23:0	RAZ.

### APIC320 LVT Timer

Bits	Description
31:18	RAZ.
17	<b>Mode.</b> Read-write. Reset: 0. 0=One-shot. 1=Periodic.
16	<b>Mask.</b> Read-write. Reset: 1. 0=Not masked. 1=Masked.
15:13	RAZ.
12	<b>DS: interrupt delivery status.</b> Read-only; updated-by-hardware. Reset: 0. 0=Idle. 1=Send pending. (Indicates that the interrupt has not yet been accepted by the core.)
11	RAZ.

10:8	<b>MsgType: message type.</b> Read-write. Reset: 000b. See 2.4.8.1.14 [Generalized Local Vector Table].
7:0	<b>Vector.</b> Read-write. Reset: 00h. Interrupt vector number.

### APIC330 LVT Thermal Sensor

Interrupts for this local vector table are caused by changes in [MSRC001\\_0061](#)[CurPstateLimit] due to SB-RMI or HTC.

Bits	Description
31:17	RAZ.
16	<b>Mask.</b> Read-write. Reset: 1. 0=Not masked. 1=Masked.
15:13	RAZ.
12	<b>DS: interrupt delivery status.</b> Read-only; updated-by-hardware. Reset: 0. 0=Idle. 1=Send pending. (Indicates that the interrupt has not yet been accepted by the core.)
11	RAZ.
10:8	<b>MsgType: message type.</b> Read-write. Reset: 000b. See 2.4.8.1.14 [Generalized Local Vector Table].
7:0	<b>Vector.</b> Read-write. Reset: 00h. Interrupt vector number.

### APIC340 LVT Performance Monitor

Interrupts for this local vector table are caused by overflows of:

- [MSRC001\\_00\[07:04\]](#) [Performance Event Counter (PERF\_CTR[3:0])].
- [MSRC001\\_024\[7,5,3,1\]](#) [Northbridge Performance Event Counter (NB\_PERF\_CTR[3:0])].

Bits	Description
31:17	RAZ.
16	<b>Mask.</b> Read-write. Reset: 1. 0=Not masked. 1=Masked.
15:13	RAZ.
12	<b>DS: interrupt delivery status.</b> Read-only; updated-by-hardware. Reset: 0. 0=Idle. 1=Send pending. (Indicates that the interrupt has not yet been accepted by the core.)
11	RAZ.
10:8	<b>MsgType: message type.</b> Read-write. Reset: 000b. See 2.4.8.1.14 [Generalized Local Vector Table].
7:0	<b>Vector.</b> Read-write. Reset: 00h. Interrupt vector number.

### APIC3[60:50] LVT LINT[1:0]

Table 183: Register Mapping for APIC3[60:50]

Register	Function
APIC350	LINT 0
APIC360	LINT 1

Bits	Description
31:17	RAZ.

16	<b>Mask.</b> Read-write. Reset: 1. 0=Not masked. 1=Masked.
15	<b>TM: trigger mode.</b> Read-write. Reset: 0. 0=Edge. 1=Level.
14	<b>RmtIRR.</b> Read-only; updated-by-hardware. Reset: 0. 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	Reserved.
12	<b>DS: interrupt delivery status.</b> Read-only; updated-by-hardware. Reset: 0. 0=Idle. 1=Send pending. (Indicates that the interrupt has not yet been accepted by the core.)
11	RAZ.
10:8	<b>MsgType: message type.</b> Read-write. Reset: 000b. See 2.4.8.1.14 [Generalized Local Vector Table].
7:0	<b>Vector.</b> Read-write. Reset: 00h. Interrupt vector number.

### APIC370 LVT Error

Bits	Description
31:17	RAZ.
16	<b>Mask.</b> Read-write. Reset: 1. 0=Not masked. 1=Masked.
15:13	Reserved.
12	<b>DS: interrupt delivery status.</b> Read-only; updated-by-hardware. Reset: 0. 0=Idle. 1=Send pending. (Indicates that the interrupt has not yet been accepted by the core.)
11	RAZ.
10:8	<b>MsgType: message type.</b> Read-write. Reset: 000b. See 2.4.8.1.14 [Generalized Local Vector Table].
7:0	<b>Vector.</b> Read-write. Reset: 00h. Interrupt vector number.

### APIC380 Timer Initial Count

Bits	Description
31:0	<b>Count.</b> Read-write. Reset: 0. The value copied into the current count register when the timer is loaded or reloaded.

### APIC390 Timer Current Count

Bits	Description
31:0	<b>Count.</b> Read-only. Reset: 0. The current value of the counter.

### APIC3E0 Timer Divide Configuration

The Div bits are encoded as follows:

**Table 184: Div[3,1:0] Value Table**

Div[3]	Div[1:0]	Resulting Timer Divide
0	00b	2
0	01b	4



**Table 184: Div[3,1:0] Value Table**

Div[3]	Div[1:0]	Resulting Timer Divide
0	10b	8
0	11b	16
1	00b	32
1	01b	64
1	10b	128
1	11b	1

Bits	Description
31:4	RAZ.
3	<b>Div[3]</b> . Read-write. Reset: 0. See <a href="#">Table 184</a> .
2	RAZ.
1:0	<b>Div[1:0]</b> . Read-write. Reset: 0. See <a href="#">Table 184</a> .

**APIC400 Extended APIC Feature**

Bits	Description
31:24	RAZ.
23:16	<b>ExtLvtCount: extended local vector table count</b> . Read-only. Reset: 04h. This specifies the number of extended LVT registers ( <a href="#">APIC[530:500]</a> ) in the local APIC.
15:3	RAZ.
2	<b>ExtApicIdCap: extended APIC ID capable</b> . Read-only. Reset: 1. 1=The processor is capable of supporting an 8-bit APIC ID, as controlled by <a href="#">APIC410[ExtApicIdEn]</a> .
1	<b>SeoiCap: specific end of interrupt capable</b> . Read-only. Reset: 1. 1=The <a href="#">APIC420 [Specific End Of Interrupt]</a> is present.
0	<b>IerCap: interrupt enable register capable</b> . Read-only. Reset: 1. This bit indicates that the <a href="#">APIC[4F0:480] [Interrupt Enable]</a> are present. See <a href="#">2.4.8.1.8 [Interrupt Masking]</a> .

**APIC410 Extended APIC Control**

Bits	Description
31:3	RAZ.
2	<b>ExtApicIdEn: extended APIC ID enable</b> . Read-write. Reset: 0. 1=Enable 8-bit APIC ID; <a href="#">APIC20[ApicId]</a> supports an 8-bit value; an interrupt broadcast in physical destination mode requires that the <a href="#">IntDest[7:0]=1111_1111b</a> (instead of <a href="#">xxxx_1111b</a> ); a match in physical destination mode occurs when <a href="#">(IntDest[7:0] == ApicId[7:0])</a> instead of <a href="#">(IntDest[3:0] == ApicId[3:0])</a> . If <a href="#">ExtApicIdEn=1</a> then program <a href="#">D18F0x68[ApicExtId]=1</a> and <a href="#">D18F0x68[ApicExtBrdCst]=1</a> .
1	<b>SeoiEn</b> . Read-write. Reset: 0. 1=Enable SEOI generation when a write to <a href="#">APIC420 [Specific End Of Interrupt]</a> is received.
0	<b>IerEn</b> . Read-write. Reset: 0. 1=Enable writes to the interrupt enable registers.

**APIC420 Specific End Of Interrupt**

Bits	Description
31:8	RAZ.
7:0	<b>EoiVec: end of interrupt vector.</b> Read-write. Reset: 0. 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**

Interrupt enables range is mapped as follows:

Table 185: [Register Mapping](#) for APIC[4F0:480]

Register	Function
APIC480	IntEn[31:0]
APIC490	IntEn[63:32]
APIC4A0	IntEn[95:64]
APIC4B0	IntEn[127:96]
APIC4C0	IntEn[159:128]
APIC4D0	IntEn[191:160]
APIC4E0	IntEn[223:192]
APIC4F0	IntEn[255:224]

Bits	Description
31:0	<b>InterruptEnableBits.</b> Read-write. Reset: FFFF_FFFFh. The interrupt enable bits can be used to enable each of the 256 interrupts. See above table.

**APIC[530:500] Extended Interrupt [3:0] Local Vector Table**

APIC500 provides a local vector table entry for IBS; See [D18F3x1CC](#). APIC510 provides a local vector table entry for error thresholding. The APIC[530:520] registers are unused.

Table 186: [Register Mapping](#) for APIC[530:500]

Register	Function
APIC500	Extended Interrupt 0 (IBS)
APIC510	Extended Interrupt 1 (Thresholding)
APIC520	Extended Interrupt 2
APIC530	Extended Interrupt 3

Bits	Description
31:17	RAZ.
16	<b>Mask.</b> Read-write. Reset: 1. 0=Not masked. 1=Masked.
15:13	RAZ.

12	<b>DS: interrupt delivery status.</b> Read-only; updated-by-hardware. Reset: 0. 0=Idle. 1=Send pending. (Indicates that the interrupt has not yet been accepted by the core.)
11	RAZ.
10:8	<b>MsgType: message type.</b> Read-write. Reset: 000b. See 2.4.8.1.14 [Generalized Local Vector Table].
7:0	<b>Vector.</b> Read-write. Reset: 00h. Interrupt vector number.

### 3.17 CPUID Instruction Registers

Processor feature capabilities and configuration information are provided through the CPUID instruction. The information is accessed by (1) selecting the CPUID function setting EAX and optionally ECX for some functions, (2) executing the CPUID instruction, and (3) reading the results in the EAX, EBX, ECX, and EDX registers. The syntax *CPUID FnXXXX\_XXXX\_EiX[\_xYYY]* refers to the function where EAX==X, and optionally ECX==Y, and the registers specified by EiX. EiX can be any single register such as {EAX, EBX, ECX, and EDX}, or a range of registers, such as E[C,B,A]X. Undefined function numbers return 0's in all 4 registers. See 2.4.10 [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 processor specific details about CPUID.

#### **CPUID Fn0000\_0000\_EAX Processor Vendor and Largest Standard Function Number**

Bits	Description
31:0	<b>LFuncStd: largest standard function.</b> Value: 0000_000Dh. The largest CPUID standard function input value supported by the processor implementation.

#### **CPUID Fn0000\_0000\_E[D,C,B]X Processor Vendor**

CPUID Fn0000\_0000\_E[D,C,B]X and CPUID Fn8000\_0000\_E[D,C,B]X return the same value.

**Table 187: Reset Mapping for CPUID Fn8000\_0000\_E[D,C,B]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	<b>Vendor.</b> The 12 8-bit ASCII character codes to create the string “AuthenticAMD”.

#### **CPUID Fn0000\_0001\_EAX Family, Model, Stepping Identifiers**

Also see CPUID Fn8000\_0001\_EAX [Family, Model, Stepping Identifiers].

**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]=07h, then Family[7:0]=16h.

**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	<b>ExtFamily: extended family.</b> CPUID Fn0000_0001_EAX[ExtFamily] is an alias of D18F3xFC[ExtFamily].
19:16	<b>ExtModel: extended model.</b> CPUID Fn0000_0001_EAX[ExtModel] is an alias of D18F3xFC[ExtModel].
15:12	Reserved.
11:8	<b>BaseFamily.</b> CPUID Fn0000_0001_EAX[BaseFamily] is an alias of D18F3xFC[BaseFamily].
7:4	<b>BaseModel.</b> CPUID Fn0000_0001_EAX[BaseModel] is an alias of D18F3xFC[BaseModel].
3:0	<b>Stepping.</b> CPUID Fn0000_0001_EAX[Stepping] is an alias of D18F3xFC[Stepping].

### CPUID Fn0000\_0001\_EBX LocalApicId, LogicalProcessorCount, CLFlush

Bits	Description
31:24	<b>LocalApicId: initial local APIC physical ID.</b> The initial APIC20[ApicId] value. See 2.4.4 [Processor Cores and Downcoring].
23:16	<b>LogicalProcessorCount: logical processor count.</b> Specifies the number of cores in the processor as CPUID Fn8000_0008_ECX[NC] + 1.
15:8	<b>CLFlush: CLFLUSH size in quadwords.</b> Value: 08h.
7:0	<b>8BitBrandId: 8 bit brand ID.</b> Value: 00h. Indicates that the brand ID is in CPUID Fn8000_0001_EBX.

### CPUID Fn0000\_0001\_ECX Feature Identifiers

These values can be over-written by MSRC001\_1004.

Bits	Description
31	RAZ. Reserved for use by hypervisor to indicate guest status.
30	<b>RDRAND: RDRAND instruction support.</b> Value: 0.
29	<b>F16C: half-precision convert instruction support.</b> Value: 1.
28	<b>AVX: AVX instruction support.</b> Value: 1. Value: 1.
27	<b>OSXSAVE: OS enabled support for XGETBV/XSETBV.</b> 1=The OS has enabled support for XGETBV/XSETBV instructions to query processor extended states.
26	<b>XSAVE: XSAVE (and related) instruction support.</b> Value: 1. 1=Support provided for the XSAVE, XRSTOR, XSETBV, and XGETBV instructions and the XFEATURE_ENABLED_MASK register.

25	<b>AES: AES instruction support.</b> Value: Product-specific.
24	Reserved.
23	<b>POPCNT: POPCNT instruction.</b> Value: 1.
22	<b>MOVBE: MOVBE instruction support.</b> Value: 1.
21	<b>x2APIC: x2APIC capability.</b> Value: 0.
20	<b>SSE42: SSE4.2 instruction support.</b> Value: 1.
19	<b>SSE41: SSE4.1 instruction support.</b> Value: 1.
18	Reserved.
17	<b>PCID: process context identifiers support.</b> Value: 0.
16:14	Reserved.
13	<b>CMPXCHG16B: CMPXCHG16B instruction.</b> Value: 1.
12	<b>FMA: FMA instruction support.</b> Value: 0.
11:10	Reserved.
9	<b>SSSE3: supplemental SSE3 extensions.</b> Value: 1.
8:4	Reserved.
3	<b>Monitor: Monitor/Mwait instructions.</b> Value: ~MSRC001_0015[MonMwaitDis].
2	Reserved.
1	<b>PCLMULQDQ: PCLMULQDQ instruction support.</b> Value: Product-specific.
0	<b>SSE3: SSE3 extensions.</b> Value: 1.

### CPUID Fn0000\_0001\_EDX Feature Identifiers

These values can be over-written by [MSRC001\\_1004](#).

Bits	Description
31:29	Reserved.
28	<b>HTT: hyper-threading technology.</b> Value: CPUID Fn8000_0008_ECX[NC]!=0. 1=Multi core product (CPUID Fn8000_0008_ECX[NC] != 0). 0=Single core product (CPUID Fn8000_0008_ECX[NC]==0).
27	Reserved.

26	<b>SSE2: SSE2 extensions.</b> Value: 1.
25	<b>SSE: SSE extensions.</b> Value: 1.
24	<b>FXSR: FXSAVE and FXRSTOR instructions.</b> Value: 1.
23	<b>MMX: MMX™ instructions.</b> Value: 1.
22:20	Reserved.
19	<b>CLFSH: CLFLUSH instruction.</b> Value: 1.
18	Reserved.
17	<b>PSE36: page-size extensions.</b> Value: 1.
16	<b>PAT: page attribute table.</b> Value: 1.
15	<b>CMOV: conditional move instructions, CMOV, FCOMI, FCMOV.</b> Value: 1.
14	<b>MCA: machine check architecture, MCG_CAP.</b> Value: 1.
13	<b>PGE: page global extension, CR4.PGE.</b> Value: 1.
12	<b>MTRR: memory-type range registers.</b> Value: 1.
11	<b>SysEnterSysExit: SYSENTER and SYSEXIT instructions.</b> Value: 1.
10	Reserved.
9	<b>APIC: advanced programmable interrupt controller (APIC) exists and is enabled.</b> Value: <a href="#">MSR0000_001B</a> [ApicEn].
8	<b>CMPXCHG8B: CMPXCHG8B instruction.</b> Value: 1.
7	<b>MCE: machine check exception, CR4.MCE.</b> Value: 1.
6	<b>PAE: physical-address extensions (PAE).</b> Value: 1.
5	<b>MSR: AMD model-specific registers (MSRs), with RDMSR and WRMSR instructions.</b> Value: 1.
4	<b>TSC: time stamp counter, RDTSC/RDTSCP instructions, CR4.TSD.</b> Value: 1.
3	<b>PSE: page-size extensions (4 MB pages).</b> Value: 1.
2	<b>DE: debugging extensions, IO breakpoints, CR4.DE.</b> Value: 1.
1	<b>VME: virtual-mode enhancements.</b> Value: 1.
0	<b>FPU: x87 floating point unit on-chip.</b> Value: 1.

**CPUID Fn0000\_000[4:2] Reserved**

Bits	Description
31:0	Reserved.

**CPUID Fn0000\_0005\_EAX Monitor/MWait**

Bits	Description
31:16	Reserved.
15:0	<b>MonLineSizeMin: smallest monitor-line size in bytes.</b> Value: 40h.

**CPUID Fn0000\_0005\_EBX Monitor/MWait**

Bits	Description
31:16	Reserved.
15:0	<b>MonLineSizeMax: largest monitor-line size in bytes.</b> Value: 40h.

**CPUID Fn0000\_0005\_ECX Monitor/MWait**

Bits	Description
31:2	Reserved.
1	<b>IBE: interrupt break-event.</b> Value: 1.
0	<b>EMX: enumerate MONITOR/MWAIT extensions.</b> Value: 1.

**CPUID Fn0000\_0005\_EDX Monitor/MWait**

Bits	Description
31:0	Reserved.

**CPUID Fn0000\_0006\_EAX Thermal and Power Management**

Bits	Description
31:3	Reserved.
2	<b>ARAT: always running APIC timer.</b> Value: 0. 1=Indicates support for APIC timer always running feature.
1:0	Reserved.

**CPUID Fn0000\_0006\_EBX Thermal and Power Management**

Bits	Description
31:0	Reserved.

**CPUID Fn0000\_0006\_ECX Thermal and Power Management**

Bits	Description
31:1	Reserved.
0	<b>EffFreq: effective frequency interface.</b> Value: 1. 1=Indicates presence of <a href="#">MSR0000_00E7</a> [Max Performance Frequency Clock Count (MPERF)] and <a href="#">MSR0000_00E8</a> [Actual Performance Frequency Clock Count (APERF)].

**CPUID Fn0000\_0006\_EDX Thermal and Power Management**

Bits	Description
31:0	Reserved.

**CPUID Fn0000\_0007\_EAX\_x0 Structured Extended Feature Identifiers (ECX=0)**

Bits	Description
31:0	Reserved.

**CPUID Fn0000\_0007\_EBX\_x0 Structured Extended Feature Identifiers (ECX=0)**

Bits	Description
31:11	Reserved.
10	<b>INVPCID: invalidate processor context ID.</b> Value: 0.
9	Reserved.
8	<b>BMI2: bit manipulation group 2 instruction support.</b> Value: 0.
7	<b>SMEP: supervisor mode execution protection.</b> Value: 0.
6	Reserved.
5	<b>AVX2: AVX extension support.</b> Value: 0.
4	Reserved.
3	<b>BMI1: bit manipulation group 1 instruction support.</b> Value: 1.
2:1	Reserved.
0	<b>FSGSBASE: FS and GS base read write instruction support.</b> Value: 0.

**CPUID Fn0000\_0007\_ECX\_x0 Structured Extended Feature Identifiers (ECX=0)**

Bits	Description
31:0	Reserved.



**CPUID Fn0000\_0007\_EDX\_x0 Structured Extended Feature Identifiers (ECX=0)**

Bits	Description
31:0	Reserved.

**CPUID Fn0000\_000[A:8] Reserved**

Bits	Description
31:0	Reserved.

**CPUID Fn0000\_000B Reserved**

Bits	Description
31:0	Reserved.

**CPUID Fn0000\_000C Reserved**

Bits	Description
31:0	Reserved.

**CPUID Fn0000\_000D\_EAX\_x0 Processor Extended State Enumeration (ECX=0)**

Bits	Description
31:0	<b>XFeatureSupportedMask[31:0]</b> . Value: 0000_0007h.

**CPUID Fn0000\_000D\_EBX\_x0 Processor Extended State Enumeration (ECX=0)**

Bits	Description
31:0	<p><b>XFeatureEnabledSizeMax</b>. Size in bytes of XSAVE/XRSTOR area for the currently enabled features in XCR0.  Value: 512 + 64 + (IF (XCR0[AVX]) THEN 256 ELSE 0 ENDIF)).  The components of this sum are described as follows:</p> <ul style="list-style-type: none"> <li>• 512: FPU/SSE save area (needed even if XCR0[SSE]=0)</li> <li>• 64: Header size (always needed).</li> <li>• Size of YMM area if YMM enabled.</li> </ul>

**CPUID Fn0000\_000D\_ECX\_x0 Processor Extended State Enumeration (ECX=0)**

Bits	Description
31:0	<b>XFeatureSupportedSizeMax.</b> Size in bytes of XSAVE/XRSTOR area for all features that the core supports. See XFeatureEnabledSizeMax. Value: 0000_0340h.

**CPUID Fn0000\_000D\_EDX\_x0 Processor Extended State Enumeration (ECX=0)**

Bits	Description
31:0	<b>XFeatureSupportedMask[63:32].</b> Value: 0000_0000h.

**CPUID Fn0000\_000D\_EAX\_x1 Processor Extended State Enumeration (ECX=1)**

Bits	Description
31:1	Reserved.
0	<b>XSAVEOPT: XSAVEOPT is available.</b> Value: 1.

**CPUID Fn0000\_000D\_E[D,C,B]X\_x1 Processor Extended State Enumeration (ECX=1)**

Bits	Description
31:0	Reserved.

**CPUID Fn0000\_000D\_EAX\_x2 Processor Extended State Enumeration (ECX=2)**

Bits	Description
31:0	<b>YmmSaveStateSize: YMM save state byte size.</b> Value: 0000_0100h.

**CPUID Fn0000\_000D\_EBX\_x2 Processor Extended State Enumeration (ECX=2)**

Bits	Description
31:0	<b>YmmSaveStateOffset: YMM save state byte offset.</b> Value: 0000_0240h.

**CPUID Fn0000\_000D\_ECX\_x2 Processor Extended State Enumeration (ECX=2)**

Bits	Description
31:0	Reserved.

**CPUID Fn0000\_000D\_EDX\_x2 Processor Extended State Enumeration (ECX=2)**

Bits	Description
31:0	Reserved.

For CPUID Fn0000\_000D, if ECX>2 then EAX/EBX/ECX/EDX will return 0.

**CPUID Fn8000\_0000\_EAX Largest Extended Function Number**

Bits	Description
31:0	<b>LFuncExt: largest extended function.</b> Value: 8000_001Eh. The largest CPUID extended function input value supported by the processor implementation.

**CPUID Fn8000\_0000\_E[D,C,B]X Processor Vendor**

CPUID Fn0000\_0000\_E[D,C,B]X and CPUID Fn8000\_0000\_E[D,C,B]X return the same value.

**Table 188: CPUID Fn8000\_0000\_E[B,C,D]X Value**

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	<b>Vendor.</b> The 12 8-bit ASCII character codes to create the string “AuthenticAMD”.

**CPUID Fn8000\_0001\_EAX Family, Model, Stepping Identifiers**

Also see CPUID Fn0000\_0001\_EAX [Family, Model, Stepping Identifiers].

Bits	Description
31:28	Reserved.
27:20	<b>ExtFamily: extended family.</b> CPUID Fn8000_0001_EAX[ExtFamily] is an alias of D18F3xFC[ExtFamily].
19:16	<b>ExtModel: extended model.</b> CPUID Fn8000_0001_EAX[ExtModel] is an alias of D18F3xFC[ExtModel].
15:12	Reserved.
11:8	<b>BaseFamily.</b> CPUID Fn8000_0001_EAX[BaseFamily] is an alias of D18F3xFC[BaseFamily].

7:4	<b>BaseModel.</b> <a href="#">CPUID Fn8000_0001_EAX</a> [BaseModel] is an alias of <a href="#">D18F3xFC</a> [BaseModel].
3:0	<b>Stepping.</b> <a href="#">CPUID Fn8000_0001_EAX</a> [Stepping] is an alias of <a href="#">D18F3xFC</a> [Stepping].

### **CPUID Fn8000\_0001\_EBX BrandId Identifier**

Bits	Description								
31:28	<p><b>PkgType: package type.</b> Specifies the package type. Value: 0h.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>FT3 (BGA)</td> </tr> <tr> <td>1h</td> <td>FS1b</td> </tr> <tr> <td>Fh-2h</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Description	0h	FT3 (BGA)	1h	FS1b	Fh-2h	Reserved
Bits	Description								
0h	FT3 (BGA)								
1h	FS1b								
Fh-2h	Reserved								
27:0	Reserved.								

### **CPUID Fn8000\_0001\_ECX Feature Identifiers**

These values can be over-written by [MSRC001\\_1005](#).

Bits	Description
31:30	Reserved.
29	Reserved.
28	<p><b>PerfCtrExtL2I: L2I performance counter extensions support.</b> Value: 1. Indicates support for <a href="#">MSRC001_023</a>[6,4,2,0] and <a href="#">MSRC001_023</a>[7,5,3,1].</p>
27	<p><b>PerfTsc: performance time-stamp counter supported.</b> Value: 0. Indicates support for <a href="#">MSRC001_0280</a> [Performance Time Stamp Counter (CU_PTSC)].</p>
26	<p><b>DataBreakpointExtension.</b> Value: 1. Indicates data breakpoint support for <a href="#">MSRC001_1027</a> and <a href="#">MSRC001_101</a>[B:9].</p>
25	Reserved.
24	<p><b>PerfCtrExtNB: NB performance counter extensions support.</b> Value: 1. Indicates support for <a href="#">MSRC001_024</a>[6,4,2,0] and <a href="#">MSRC001_024</a>[7,5,3,1].</p>
23	<p><b>PerfCtrExtCore: core performance counter extensions support.</b> Value: 0. Indicates support for <a href="#">MSRC001_020</a>[A,8,6,4,2,0] and <a href="#">MSRC001_020</a>[B,9,7,5,3,1].</p>
22	<p><b>TopologyExtensions: topology extensions support.</b> Value: 1. Indicates support for <a href="#">CPUID Fn8000_001D_EAX_x0</a>-<a href="#">CPUID Fn8000_001E_EDX</a>.</p>
21	<p><b>TBM: trailing bit manipulation instruction support.</b> Value: 0.</p>
20	Reserved.
19	Reserved.
18	Reserved.

17	<b>TCE: translation cache extension.</b> Value: 0.
16	<b>FMA4: 4-operand FMA instruction support.</b> Value: 0.
15	<b>LWP: lightweight profiling support.</b> Value: 0.
14	Reserved.
13	<b>WDT: watchdog timer support.</b> Value: 1.
12	<b>SKINIT: SKINIT and STGI support.</b> Value: 1.
11	<b>XOP: extended operation support.</b> Value: 0.
10	<b>IBS: Instruction Based Sampling.</b> Value: 1.
9	<b>OSVW: OS Visible Work-around support.</b> Value: 1.
8	<b>3DNowPrefetch: Prefetch and PrefetchW instructions.</b> Value: 1.
7	<b>MisAlignSse: Misaligned SSE Mode.</b> Value: 1.
6	<b>SSE4A: EXTRQ, INSERTQ, MOVNTSS, and MOVNTSD instruction support.</b> Value: 1.
5	<b>ABM: advanced bit manipulation.</b> Value: 1. LZCNT instruction support.
4	<b>AltMovCr8: LOCK MOV CR0 means MOV CR8.</b> Value: 1.
3	<b>ExtApicSpace: extended APIC register space.</b> Value: 1.
2	<b>SVM: Secure Virtual Mode feature.</b> Value: Product-specific. Indicates support for: VMRUN, VMLOAD, VMSAVE, CLGI, VMMCALL, and INVLPGA.
1	<b>CmpLegacy: core multi-processing legacy mode.</b> Value: Product-specific. 1=Multi core product (CPUID Fn8000_0008_ECX[NC] != 0). 0=Single core product (CPUID Fn8000_0008_ECX[NC] == 0).
0	<b>LahfSahf: LAHF/SAHF instructions.</b> Value: 1.

### CPUID Fn8000\_0001\_EDX Feature Identifiers

These values can be over-written by [MSRC001\\_1005](#).

Bits	Description
31	<b>3DNow: 3DNow!™ instructions.</b> Value: 0.
30	<b>3DNowExt: AMD extensions to 3DNow!™ instructions.</b> Value: 0.
29	<b>LM: long mode.</b> Value: 1.
28	Reserved.
27	<b>RDTSCP: RDTSCP instruction.</b> Value: 1.

26	<b>Page1GB: one GB large page support.</b> Value: 1.
25	<b>FFXSR: FXSAVE and FXRSTOR instruction optimizations.</b> Value: 1.
24	<b>FXSR: FXSAVE and FXRSTOR instructions.</b> Value: 1.
23	<b>MMX: MMX™ instructions.</b> Value: 1.
22	<b>MmxExt: AMD extensions to MMX™ instructions.</b> Value: 1.
21	Reserved.
20	<b>NX: no-execute page protection.</b> Value: 1.
19:18	Reserved.
17	<b>PSE36: page-size extensions.</b> Value: 1.
16	<b>PAT: page attribute table.</b> Value: 1.
15	<b>CMOV: conditional move instructions, CMOV, FCOMI, FCMOV.</b> Value: 1.
14	<b>MCA: machine check architecture, MCG_CAP.</b> Value: 1.
13	<b>PGE: page global extension, CR4.PGE.</b> Value: 1.
12	<b>MTRR: memory-type range registers.</b> Value: 1.
11	<b>SysCallSysRet: SYSCALL and SYSRET instructions.</b> Value: 1.
10	Reserved.
9	<b>APIC: advanced programmable interrupt controller (APIC) exists and is enabled.</b> Value: MSR0000_001B[ApicEn].
8	<b>CMPXCHG8B: CMPXCHG8B instruction.</b> Value: 1.
7	<b>MCE: machine check exception, CR4.MCE.</b> Value: 1.
6	<b>PAE: physical-address extensions (PAE).</b> Value: 1.
5	<b>MSR: model-specific registers (MSRs), with RDMSR and WRMSR instructions.</b> Value: 1.
4	<b>TSC: time stamp counter, RDTSC/RDTSCP instructions, CR4.TSD.</b> Value: 1.
3	<b>PSE: page-size extensions (4 MB pages).</b> Value: 1.
2	<b>DE: debugging extensions, IO breakpoints, CR4.DE.</b> Value: 1.
1	<b>VME: virtual-mode enhancements.</b> Value: 1.
0	<b>FPU: x87 floating point unit on-chip.</b> Value: 1.

**CPUID Fn8000\_000[4:2]\_E[D,C,B,A]X Processor Name String Identifier**

Table 189: Valid Values for CPUID Fn8000\_000[4:2]\_E[D,C,B,A]X

Register	Value
CPUID Fn8000_0002_EAX	MSRC001_0030[31:0]
CPUID Fn8000_0002_EBX	MSRC001_0030[63:32]
CPUID Fn8000_0002_ECX	MSRC001_0031[31:0]
CPUID Fn8000_0002_EDX	MSRC001_0031[63:32]
CPUID Fn8000_0003_EAX	MSRC001_0032[31:0]
CPUID Fn8000_0003_EBX	MSRC001_0032[63:32]
CPUID Fn8000_0003_ECX	MSRC001_0033[31:0]
CPUID Fn8000_0003_EDX	MSRC001_0033[63:32]

Table 189: Valid Values for [CPUID Fn8000\\_000\[4:2\]\\_E\[D,C,B,A\]X](#)

CPUID Fn8000_0004_EAX	MSRC001_0034[31:0]
CPUID Fn8000_0004_EBX	MSRC001_0034[63:32]
CPUID Fn8000_0004_ECX	MSRC001_0035[31:0]
CPUID Fn8000_0004_EDX	MSRC001_0035[63:32]

Bits	Description
31:0	<b>ProcName: processor name.</b> These return the ASCII string corresponding to the processor name, stored in <a href="#">MSRC001_00[35:30]</a> [ <a href="#">Processor Name String</a> ].

### **CPUID Fn8000\_0005\_EAX L1 TLB 2M/4M Identifiers**

This function provides the processor's first level cache and TLB characteristics for each core.

Bits	Description
31:24	<b>L1DTlb2and4MAssoc: data TLB associativity for 2 MB and 4 MB pages.</b> Value: FFh. See: <a href="#">CPUID Fn8000_0005_ECX</a> [L1DcAssoc].
23:16	<b>L1DTlb2and4MSize: data TLB number of entries for 2 MB and 4 MB pages.</b> Value: 8. 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	<b>L1ITlb2and4MAssoc: instruction TLB associativity for 2 MB and 4 MB pages.</b> Value: FFh. See: <a href="#">CPUID Fn8000_0005_ECX</a> [L1DcAssoc].
7:0	<b>L1ITlb2and4MSize: instruction TLB number of entries for 2 MB and 4 MB pages.</b> Value: 8. 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 4K Identifiers**

See: [CPUID Fn8000\\_0005\\_EAX](#).

Bits	Description
31:24	<b>L1DTlb4KAssoc: data TLB associativity for 4 KB pages.</b> Value: FFh. See: <a href="#">CPUID Fn8000_0005_ECX</a> [L1DcAssoc].
23:16	<b>L1DTlb4KSize: data TLB number of entries for 4 KB pages.</b> Value: 40.
15:8	<b>L1ITlb4KAssoc: instruction TLB associativity for 4 KB pages.</b> Value: FFh. ITLB associativity for 4 KB pages is reported by <a href="#">CPUID Fn8000_0006_EBX</a> [L2ITlb4KAssoc].
7:0	<b>L1ITlb4KSize: instruction TLB number of entries for 4 KB pages.</b> Value: 32. ITLB size for 4 KB pages is reported by <a href="#">CPUID Fn8000_0006_EBX</a> [L2ITlb4KSize].

**CPUID Fn8000\_0005\_ECX L1 Data Cache Identifiers**

This function provides first level cache characteristics for each core.

Bits	Description														
31:24	<b>L1DcSize: L1 data cache size in KB.</b> Value: 32.														
23:16	<b>L1DcAssoc: L1 data cache associativity.</b> Value: 8. <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00h</td> <td>Reserved</td> </tr> <tr> <td>01h</td> <td>1 way (direct mapped)</td> </tr> <tr> <td>02h</td> <td>2 way</td> </tr> <tr> <td>03h</td> <td>3 way</td> </tr> <tr> <td>FEh-04h</td> <td>[L1IcAssoc] way</td> </tr> <tr> <td>FFh</td> <td>Fully associative</td> </tr> </tbody> </table>	Bits	Description	00h	Reserved	01h	1 way (direct mapped)	02h	2 way	03h	3 way	FEh-04h	[L1IcAssoc] way	FFh	Fully associative
Bits	Description														
00h	Reserved														
01h	1 way (direct mapped)														
02h	2 way														
03h	3 way														
FEh-04h	[L1IcAssoc] way														
FFh	Fully associative														
15:8	<b>L1DcLinesPerTag: L1 data cache lines per tag.</b> Value: 1.														
7:0	<b>L1DcLineSize: L1 data cache line size in bytes.</b> Value: 64.														

**CPUID Fn8000\_0005\_EDX L1 Instruction Cache Identifiers**

This function provides first level cache characteristics for each core.

Bits	Description
31:24	<b>L1IcSize: L1 instruction cache size KB.</b> Value: 32.
23:16	<b>L1IcAssoc: L1 instruction cache associativity.</b> Value: 2. See: <a href="#">CPUID Fn8000_0005_ECX[L1DcAssoc]</a> .
15:8	<b>L1IcLinesPerTag: L1 instruction cache lines per tag.</b> Value: 1.
7:0	<b>L1IcLineSize: L1 instruction cache line size in bytes.</b> Value: 64.

**CPUID Fn8000\_0006\_EAX L2 TLB 2M/4M Identifiers**

This function provides the processor's second level cache and TLB characteristics for each core.

Bits	Description
31:28	<b>L2DTlb2and4MAssoc: L2 data TLB associativity for 2 MB and 4 MB pages.</b> Value: 2. See: <a href="#">CPUID Fn8000_0006_ECX[L2Assoc]</a> .
27:16	<b>L2DTlb2and4MSize: L2 data TLB number of entries for 2 MB and 4 MB pages.</b> Value: 256. 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	<b>L2ITlb2and4MAssoc: L2 instruction TLB associativity for 2 MB and 4 MB pages.</b> Value: 0. See: <a href="#">CPUID Fn8000_0006_ECX[L2Assoc]</a> .
11:0	<b>L2ITlb2and4MSize: L2 instruction TLB number of entries for 2 MB and 4 MB pages.</b> Value: 0. 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\_0006\_EBX L2 TLB 4K Identifiers**

This function provides second level TLB characteristics for 4K pages shared by each core on a L2 complex.

Bits	Description
31:28	<b>L2DTlb4KAssoc: L2 data TLB associativity for 4 KB pages.</b> Value: 4. See: <a href="#">CPUID Fn8000_0006_ECX[L2Assoc]</a> .
27:16	<b>L2DTlb4KSize: L2 data TLB number of entries for 4 KB pages.</b> Value: 512.
15:12	<b>L2ITlb4KAssoc: L2 instruction TLB associativity for 4 KB pages.</b> Value: 4. See: <a href="#">CPUID Fn8000_0006_ECX[L2Assoc]</a> .
11:0	<b>L2ITlb4KSize: L2 instruction TLB number of entries for 4 KB pages.</b> Value: 512.

**CPUID Fn8000\_0006\_ECX L2 Cache Identifiers**

Bits	Description																																				
31:16	<b>L2Size: L2 cache size in KB.</b> Value: Product-specific. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>03FFh-0000h</td> <td>Reserved</td> </tr> <tr> <td>0400h</td> <td>1 MB</td> </tr> <tr> <td>07FFh-0401h</td> <td>Reserved</td> </tr> <tr> <td>0800h</td> <td>2 MB</td> </tr> <tr> <td>FFFFh-0801h</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Description	03FFh-0000h	Reserved	0400h	1 MB	07FFh-0401h	Reserved	0800h	2 MB	FFFFh-0801h	Reserved																								
Bits	Description																																				
03FFh-0000h	Reserved																																				
0400h	1 MB																																				
07FFh-0401h	Reserved																																				
0800h	2 MB																																				
FFFFh-0801h	Reserved																																				
15:12	<b>L2Assoc: L2 cache associativity.</b> Value: 8. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Bits</th> <th>Description</th> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Disabled.</td> <td>8h</td> <td>16 ways</td> </tr> <tr> <td>1h</td> <td>1 way (direct mapped)</td> <td>9h</td> <td>Reserved</td> </tr> <tr> <td>2h</td> <td>2 ways</td> <td>Ah</td> <td>32 ways</td> </tr> <tr> <td>3h</td> <td>Reserved</td> <td>Bh</td> <td>48 ways</td> </tr> <tr> <td>4h</td> <td>4 ways</td> <td>Ch</td> <td>64 ways</td> </tr> <tr> <td>5h</td> <td>Reserved</td> <td>Dh</td> <td>96 ways</td> </tr> <tr> <td>6h</td> <td>8 ways</td> <td>Eh</td> <td>128 ways</td> </tr> <tr> <td>7h</td> <td>Reserved</td> <td>Fh</td> <td>Fully associative</td> </tr> </tbody> </table>	Bits	Description	Bits	Description	0h	Disabled.	8h	16 ways	1h	1 way (direct mapped)	9h	Reserved	2h	2 ways	Ah	32 ways	3h	Reserved	Bh	48 ways	4h	4 ways	Ch	64 ways	5h	Reserved	Dh	96 ways	6h	8 ways	Eh	128 ways	7h	Reserved	Fh	Fully associative
Bits	Description	Bits	Description																																		
0h	Disabled.	8h	16 ways																																		
1h	1 way (direct mapped)	9h	Reserved																																		
2h	2 ways	Ah	32 ways																																		
3h	Reserved	Bh	48 ways																																		
4h	4 ways	Ch	64 ways																																		
5h	Reserved	Dh	96 ways																																		
6h	8 ways	Eh	128 ways																																		
7h	Reserved	Fh	Fully associative																																		

11:8	<b>L2LinesPerTag: L2 cache lines per tag.</b> Value: 1.
7:0	<b>L2LineSize: L2 cache line size in bytes.</b> Value: 64.

### CPUID Fn8000\_0006\_EDX L3 Cache Identifiers

This function provides third level cache characteristics shared by all cores.

Bits	Description
31:18	<b>L3Size: L3 cache size.</b> Value: 0.
17:16	Reserved.
15:12	<b>L3Assoc: L3 cache associativity.</b> Value: 0.
11:8	<b>L3LinesPerTag: L3 cache lines per tag.</b> Value: 0.
7:0	<b>L3LineSize: L3 cache line size in bytes.</b> Value: 0.

### CPUID Fn8000\_0007\_EAX Processor Feedback Capabilities

Bits	Description								
31:16	<b>MaxWrapTime.</b> Value: 0D6Bh. (3435 ms) Specifies the maximum time between reads that software should use to avoid two wraps. A read of at least once every MaxWrapTime seconds will result in either zero or one wrap during that interval.								
15:8	<b>Version.</b> Value: 01h. Specifies the processor feedback capabilities version.								
7:0	<b>NumberOfMonitors.</b> Value: 01h. Specifies the number of processor feedback MSR pairs supported, from <a href="#">MSRC001_0080</a> to <a href="#">MSRC001_008</a> [0+((NumberOfMonitors-1)*2)+1]. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00h</td> <td>Reserved.</td> </tr> <tr> <td>01h</td> <td><a href="#">MSRC001_0080</a> to <a href="#">MSRC001_0081</a> supported.</td> </tr> <tr> <td>FFh-02h</td> <td>Reserved.</td> </tr> </tbody> </table>	Bits	Description	00h	Reserved.	01h	<a href="#">MSRC001_0080</a> to <a href="#">MSRC001_0081</a> supported.	FFh-02h	Reserved.
Bits	Description								
00h	Reserved.								
01h	<a href="#">MSRC001_0080</a> to <a href="#">MSRC001_0081</a> supported.								
FFh-02h	Reserved.								

### CPUID Fn8000\_0007\_EBX RAS Capabilities

Bits	Description
31:3	Reserved.
2	<b>HWA: hardware assert supported.</b> 1=Indicates support for <a href="#">MSRC001_10[DF:C0]</a> . Value: 0.
1	<b>SUCCOR: Software uncorrectable error containment and recovery capability.</b> Value: 0.
0	<b>McaOverflowRecov: MCA overflow recovery support.</b> Value: 1. 1=MCA overflow conditions ( <a href="#">MCi_STATUS[Overflow]=1</a> ) are not fatal; software may safely ignore such conditions. 0=MCA overflow conditions require software to shut down the system. See <a href="#">2.14.1.6 [Handling Machine Check Exceptions]</a> .

**CPUID Fn8000\_0007\_ECX 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
31:13	Reserved.
12	Reserved.
11	<b>ProcFeedbackInterface: processor feedback interface.</b> Value: 1. 1=Indicates support for processor feedback interface; <a href="#">CPUID Fn8000_0007_EAX</a> .
10	<b>EffFreqRO: read-only effective frequency interface.</b> Value: 0. 1=Indicates presence of MSRC000_00E7 [Read-Only Max Performance Frequency Clock Count (MPerfReadOnly)] and MSRC000_00E8 [Read-Only Actual Performance Frequency Clock Count (APerfReadOnly)].
9	<b>CPB: core performance boost.</b> Value: Product-specific. 1= Indicates presence of <a href="#">MSRC001_0015</a> [CpbDis] and support for core performance boost. See <a href="#">2.5.9 [Application Power Management (APM)]</a> .
8	<b>TscInvariant: TSC invariant.</b> Value: 1. The TSC rate is invariant.
7	<b>HwpState: hardware P-state control.</b> Value: 1. <a href="#">MSRC001_0061</a> [P-state Current Limit], <a href="#">MSRC001_0062</a> [P-state Control] and <a href="#">MSRC001_0063</a> [P-state Status] exist.
6	<b>100MHzSteps: 100 MHz multiplier Control.</b> Value: 1.
5	Reserved.
4	<b>TM: hardware thermal control (HTC).</b> Value: Product-specific.
3	<b>TTP: THERMTRIP.</b> Value: 1.
2	<b>VID: Voltage ID control.</b> Value: 0. Function replaced by HwpState.
1	<b>FID: Frequency ID control.</b> Value: 0. Function replaced by HwpState.
0	<b>TS: Temperature sensor.</b> Value: 1.

**CPUID Fn8000\_0008\_EAX Long Mode Address Size Identifiers**

This provides information about the maximum physical and linear address width supported by the processor.

Bits	Description
31:24	Reserved.
23:16	<b>GuestPhysAddrSize: maximum guest physical byte address size in bits.</b> Value: 0. 0=The maximum guest physical address size defined by PhysAddrSize.

15:8	<b>LinAddrSize: Maximum linear byte address size in bits.</b> Value: IF (CUID Fn8000_0001_EDX[LM]) THEN 30h ELSE 20h ENDIF.
7:0	<b>PhysAddrSize: Maximum physical byte address size in bits.</b> Value: 28h.

**CPUID Fn8000\_0008\_EBX Reserved**

Bits	Description
31:0	Reserved.

**CPUID Fn8000\_0008\_ECX Size Identifiers**

This provides information about the number of cores supported by the processor.

Bits	Description										
31:18	Reserved.										
17:16	<b>PerfTscSize: performance time-stamp counter size.</b> Value: 00b. Indicates the size of MSRC001_0280[PTSC]. Valid only when (CUID Fn8000_0001_ECX[PerfTsc]==1). <table border="1" data-bbox="267 955 665 1129"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>40 bits</td> </tr> <tr> <td>01b</td> <td>48 bits</td> </tr> <tr> <td>10b</td> <td>56 bits</td> </tr> <tr> <td>11b</td> <td>64 bits</td> </tr> </tbody> </table>	Bits	Description	00b	40 bits	01b	48 bits	10b	56 bits	11b	64 bits
Bits	Description										
00b	40 bits										
01b	48 bits										
10b	56 bits										
11b	64 bits										
15:12	<b>ApicIdCoreIdSize: APIC ID size.</b> Value: 3h. The number of bits in the initial APIC20[ApicId] value that indicate core ID within a processor.										
11:8	Reserved.										
7:0	<b>NC: number of cores - 1.</b> The number of cores in the processor is NC+1 (e.g., if NC=0, then there is one core). See 2.4.4 [Processor Cores and Downcoring]. Value: D18F5x84[CmpCap]-NumO-nes(D18F3x190[DisCore]).										

**CPUID Fn8000\_0008\_EDX Reserved**

Bits	Description
31:0	Reserved.

**CPUID Fn8000\_0009 Reserved**

Bits	Description
31:0	Reserved.

**CPUID Fn8000\_000A\_EAX SVM Revision and Feature Identification**

This provides SVM revision. If (CPUID Fn8000\_0001\_ECX[SVM]==0) then CPUID Fn8000\_000A\_EAX is reserved.

Bits	Description
31:8	Reserved.
7:0	<b>SvmRev: SVM revision.</b> Value: 01h.

**CPUID Fn8000\_000A\_EBX SVM Revision and Feature Identification**

This provides SVM revision and feature information. If (CPUID Fn8000\_0001\_ECX[SVM]==0) then CPUID Fn8000\_000A\_EBX is reserved.

Bits	Description
31:0	<b>NASID: number of address space identifiers (ASID).</b> Value: 8h.

**CPUID Fn8000\_000A\_ECX SVM Revision and Feature Identification**

Bits	Description
31:0	Reserved.

**CPUID Fn8000\_000A\_EDX SVM Revision and Feature Identification**

This provides SVM feature information. If (CPUID Fn8000\_0001\_ECX[SVM]==0) then CPUID Fn8000\_000A\_EDX is reserved.

Bits	Description
31:14	Reserved.
13	<b>AVIC: AMD virtual interrupt controller.</b> Value: 0. 1=Support indicated for SVM mode virtualized interrupt controller; Indicates support for MSRC001_011B [AVIC Doorbell].
12	<b>PauseFilterThreshold: PAUSE filter threshold.</b> Value: 1.
11	Reserved.
10	<b>PauseFilter: pause intercept filter.</b> Value: 1.
9:8	Reserved.
7	<b>DecodeAssists: decode assists.</b> Value: 1.
6	<b>FlushByAsid: flush by ASID.</b> Value: 1.
5	<b>VmcbClean: VMCB clean bits.</b> Value: 0.

4	<b>TscRateMsr: MSR based TSC rate control.</b> Value: 1. 1=Indicates support for TSC ratio <a href="#">MSRC000_0104</a> .
3	<b>NRIPS: NRIP Save.</b> Value: 1.
2	<b>SVML: SVM lock.</b> Value: 1.
1	<b>LbrVirt: LBR virtualization.</b> Value: 1.
0	<b>NP: nested paging.</b> Value: 1.

### CPUID Fn8000\_00[18:0B] Reserved

Bits	Description
31:0	Reserved.

### CPUID Fn8000\_0019\_EAX L1 TLB 1G Identifiers

This function provides first level TLB characteristics for 1G pages shared by each core on a L2 complex.

Bits	Description
31:28	<b>L1DTlb1GAssoc: L1 data TLB associativity for 1 GB pages.</b> See: <a href="#">CPUID Fn8000_0006_ECX[L2Assoc]</a> . Value: 0.
27:16	<b>L1DTlb1GSize: L1 data TLB number of entries for 1 GB pages.</b> Value: 0.
15:12	<b>L1ITlb1GAssoc: L1 instruction TLB associativity for 1 GB pages.</b> See: <a href="#">CPUID Fn8000_0006_ECX[L2Assoc]</a> . Value: 0.
11:0	<b>L1ITlb1GSize: L1 instruction TLB number of entries for 1 GB pages.</b> Value: 0.

### CPUID Fn8000\_0019\_EBX L2 TLB 1G 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	<b>L2DTlb1GAssoc: L2 data TLB associativity for 1 GB pages.</b> See: <a href="#">CPUID Fn8000_0006_ECX[L2Assoc]</a> . Value: 0.
27:16	<b>L2DTlb1GSize: L2 data TLB number of entries for 1 GB pages.</b> Value: 0.

15:12	<b>L2ITlb1GAssoc: L2 instruction TLB associativity for 1 GB pages.</b> See: <a href="#">CPUID Fn8000_0006_ECX[L2Assoc]</a> . Value: 0.
11:0	<b>L2ITlb1GSize: L2 instruction TLB number of entries for 1 GB pages.</b> Value: 0.

**CPUID Fn8000\_0019\_E[D,C]X Reserved**

Bits	Description
31:0	Reserved.

**CPUID Fn8000\_001A\_EAX Performance Optimization Identifiers**

This function returns performance related information. For more details on how to use these bits to optimize software, see the optimization guide.

Bits	Description
31:3	Reserved.
2	<b>FP256.</b> Value: 0.
1	<b>MOVU.</b> Value: 1.
0	<b>FP128.</b> Value: 1.

**CPUID Fn8000\_001A\_E[D,C,B]X Reserved**

Bits	Description
31:0	Reserved.

**CPUID Fn8000\_001B\_EAX Instruction Based Sampling Identifiers**

This function returns IBS feature information.

Bits	Description
31:11	Reserved.
10	<b>IbsOpData4: IBS op data 4 MSR supported.</b> Value: 0. See MSRC001_103D [IBS Op Data 4 (DC_IBS_DATA2)].
9	<b>IbsFetchCtlExtd: IBS fetch control extended MSR supported.</b> Value: 0. 1=Indicates support for MSRC001_103C [IBS Fetch Control Extended (IC_IBS_EXTD_CTL)].

8	<b>OpBrnFuse: fused branch micro-op indication supported.</b> Value: 0. 1=Indicates support for <a href="#">MSRC001_1035</a> [IbsOpBrnFuse].
7	<b>RipInvalidChk: invalid RIP indication supported.</b> Value: 1. 1=Indicates support for <a href="#">MSRC001_1035</a> [IbsRipInvalid].
6	<b>OpCntExt: IbsOpCurCnt and IbsOpMaxCnt extend by 7 bits.</b> Value: 1. 1=Indicates support for <a href="#">MSRC001_1033</a> [IbsOpCurCnt[26:20], IbsOpMaxCnt[26:20]].
5	<b>BrnTrgt: branch target address reporting supported.</b> Value: 1.
4	<b>OpCnt: op counting mode supported.</b> Value: 1.
3	<b>RdWrOpCnt: read write of op counter supported.</b> Value: 1.
2	<b>OpSam: IBS execution sampling supported.</b> Value: 1.
1	<b>FetchSam: IBS fetch sampling supported.</b> Value: 1.
0	<b>IBSFFV: IBS feature flags valid.</b> Value: 1.

### **CPUID Fn8000\_001B\_E[D,C,B]X Instruction Based Sampling Identifiers**

Bits	Description
31:0	Reserved.

### **CPUID Fn8000\_001C\_EAX Lightweight Profiling Capabilities 0**

This function returns IBS feature information; see the Lightweight Profiling Specification section titled “Detecting LWP”. If (CPUID Fn8000\_0001\_ECX[LWP]==0) then CPUID Fn8000\_001C\_E[D,C,B,A]X is reserved.

Bits	Description
31:0	Reserved.

### **CPUID Fn8000\_001C\_EBX Lightweight Profiling Capabilities 0**

See [CPUID Fn8000\\_001C\\_EAX](#).

Bits	Description
31:0	Reserved.

### **CPUID Fn8000\_001C\_ECX Lightweight Profiling Capabilities 0**

See [CPUID Fn8000\\_001C\\_EAX](#).

Bits	Description
31:0	Reserved.

### **CPUID Fn8000\_001C\_EDX Lightweight Profiling Capabilities 0**

See [CPUID Fn8000\\_001C\\_EAX](#).



Bits	Description
31:0	Reserved.

### CPUID Fn8000\_001D\_EAX\_x0 Cache Properties

CPUID Fn8000\_001D\_EAX\_x0 reports topology information for the DC.

If (CPUID Fn8000\_0001\_ECX[TopologyExtensions]==0) then CPUID Fn8000\_001D\_E[D,C,B,A]X is reserved.

Table 190: ECX mapping to Cache Type for CPUID Fn8000\_001D\_E[D,C,B,A]X

ECX	Cache Type
0	DC
1	IC
2	L2
3	Null

Bits	Description												
31:26	Reserved.												
25:14	<b>NumSharingCache: number of cores sharing cache.</b> Value: 000h. The number of cores sharing this cache is NumSharingCache+1.												
13:10	Reserved.												
9	<b>FullyAssociative: fully associative cache.</b> Value: 0. 1=Cache is fully associative.												
8	<b>SelfInitialization: cache is self-initializing.</b> Value: 1. 1=Cache is self initializing; cache does not need software initialization.												
7:5	<b>CacheLevel: cache level.</b> Identifies the cache level. Value: 001b. <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>Reserved.</td> </tr> <tr> <td>001b</td> <td>Level 1</td> </tr> <tr> <td>010b</td> <td>Level 2</td> </tr> <tr> <td>011b</td> <td>Level 3</td> </tr> <tr> <td>111b-100b</td> <td>Reserved.</td> </tr> </tbody> </table>	Bits	Description	000b	Reserved.	001b	Level 1	010b	Level 2	011b	Level 3	111b-100b	Reserved.
Bits	Description												
000b	Reserved.												
001b	Level 1												
010b	Level 2												
011b	Level 3												
111b-100b	Reserved.												
4:0	<b>CacheType: cache type.</b> Identifies the type of cache. Value: 01h. <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00h</td> <td>Null; no more caches.</td> </tr> <tr> <td>01h</td> <td>Data cache</td> </tr> <tr> <td>02h</td> <td>Instruction cache</td> </tr> <tr> <td>03h</td> <td>Unified cache</td> </tr> <tr> <td>1Fh-04h</td> <td>Reserved.</td> </tr> </tbody> </table>	Bits	Description	00h	Null; no more caches.	01h	Data cache	02h	Instruction cache	03h	Unified cache	1Fh-04h	Reserved.
Bits	Description												
00h	Null; no more caches.												
01h	Data cache												
02h	Instruction cache												
03h	Unified cache												
1Fh-04h	Reserved.												

### CPUID Fn8000\_001D\_EAX\_x1 Cache Properties

CPUID Fn8000\_001D\_EAX\_x1 reports topology information for the IC. See CPUID Fn8000\_001D\_EAX\_x0.

Bits	Description
31:26	Reserved.
25:14	<b>NumSharingCache: number of cores sharing cache.</b> See: <a href="#">CPUID Fn8000_001D_EAX_x0[NumSharingCache]</a> . Value: 000h.
13:10	Reserved.
9	<b>FullyAssociative: fully associative cache.</b> Value: 0. See: <a href="#">CPUID Fn8000_001D_EAX_x0[FullyAssociative]</a> .
8	<b>SelfInitialization: cache is self-initializing.</b> Value: 1. See: <a href="#">CPUID Fn8000_001D_EAX_x0[SelfInitialization]</a> .
7:5	<b>CacheLevel: cache level.</b> Identifies the cache level. Value: 001b. See: <a href="#">CPUID Fn8000_001D_EAX_x0[CacheLevel]</a> .
4:0	<b>CacheType: cache type.</b> Value: 02h. See: <a href="#">CPUID Fn8000_001D_EAX_x0[CacheType]</a> .

#### **CPUID Fn8000\_001D\_EAX\_x2 Cache Properties**

[CPUID Fn8000\\_001D\\_EAX\\_x2](#) reports topology information for the L2. See [CPUID Fn8000\\_001D\\_EAX\\_x0](#).

Bits	Description
31:26	Reserved.
25:14	<b>NumSharingCache: number of cores sharing cache.</b> See: <a href="#">CPUID Fn8000_001D_EAX_x0[NumSharingCache]</a> . Value: <a href="#">CPUID Fn8000_001E_EBX[ThreadsPerComputeUnit]</a> .
13:10	Reserved.
9	<b>FullyAssociative: fully associative cache.</b> Value: 0. See: <a href="#">CPUID Fn8000_001D_EAX_x0[FullyAssociative]</a> .
8	<b>SelfInitialization: cache is self-initializing.</b> Value: 1. See: <a href="#">CPUID Fn8000_001D_EAX_x0[SelfInitialization]</a> .
7:5	<b>CacheLevel: cache level.</b> Identifies the cache level. Value: 010b. See: <a href="#">CPUID Fn8000_001D_EAX_x0[CacheLevel]</a> .
4:0	<b>CacheType: cache type.</b> Value: 03h. See: <a href="#">CPUID Fn8000_001D_EAX_x0[CacheType]</a> .

#### **CPUID Fn8000\_001D\_EAX\_x3 Cache Properties**

[CPUID Fn8000\\_001D\\_EAX\\_x3](#) reports done/null. See [CPUID Fn8000\\_001D\\_EAX\\_x0](#).

Bits	Description
31:5	Reserved.
4:0	<b>CacheType: cache type.</b> Value: 00h. See: <a href="#">CPUID Fn8000_001D_EAX_x0[CacheType]</a> .

**CPUID Fn8000\_001D\_EBX\_x0 Cache Properties**

CPUID Fn8000\_001D\_EBX\_x0 reports topology information for the DC. See CPUID Fn8000\_001D\_EAX\_x0.

Bits	Description
31:22	<b>CacheNumWays: cache number of ways.</b> Cache number of ways is CacheNumWays+1. Value: 007h.
21:12	<b>CachePhysPartitions: cache physical line partitions.</b> Value: 000h. Cache partitions is CachePhysPartitions+1.
11:0	<b>CacheLineSize: cache line size in bytes.</b> Value: 03Fh. Cache line size in bytes is CacheLineSize+1.

**CPUID Fn8000\_001D\_EBX\_x1 Cache Properties**

CPUID Fn8000\_001D\_EBX\_x1 reports topology information for the IC. See CPUID Fn8000\_001D\_EAX\_x0.

Bits	Description
31:22	<b>CacheNumWays: cache number of ways.</b> See: CPUID Fn8000_001D_EBX_x0[CacheNumWays]. Value: 001h.
21:12	<b>CachePhysPartitions: cache physical line partitions.</b> Value: 000h. See: CPUID Fn8000_001D_EBX_x0[CachePhysPartitions].
11:0	<b>CacheLineSize: cache line size in bytes.</b> Value: 03Fh. See: CPUID Fn8000_001D_EBX_x0[CacheLineSize].

**CPUID Fn8000\_001D\_EBX\_x2 Cache Properties**

CPUID Fn8000\_001D\_EBX\_x2 reports topology information for the L2. See CPUID Fn8000\_001D\_EAX\_x0.

Bits	Description
31:22	<b>CacheNumWays: cache number of ways.</b> Value: 00Fh. See: CPUID Fn8000_001D_EBX_x0[CacheNumWays].
21:12	<b>CachePhysPartitions: cache physical line partitions.</b> Value: 000h. See: CPUID Fn8000_001D_EBX_x0[CachePhysPartitions].
11:0	<b>CacheLineSize: cache line size in bytes.</b> Value: 03Fh. See: CPUID Fn8000_001D_EBX_x0[CacheLineSize].

**CPUID Fn8000\_001D\_EBX\_x3 Cache Properties**

CPUID Fn8000\_001D\_EAX\_x3 reports done/null. See CPUID Fn8000\_001D\_EAX\_x0.

Bits	Description
31:22	<b>CacheNumWays: cache number of ways.</b> Value: 0. See: <a href="#">CPUID Fn8000_001D_EBX_x0</a> [CacheNumWays].
21:12	<b>CachePhysPartitions: cache physical line partitions.</b> Value: 000h. See: <a href="#">CPUID Fn8000_001D_EBX_x0</a> [CachePhysPartitions].
11:0	<b>CacheLineSize: cache line size in bytes.</b> Value: 0. See: <a href="#">CPUID Fn8000_001D_EBX_x0</a> [CacheLineSize].

### **CPUID Fn8000\_001D\_ECX\_x0 Cache Properties**

[CPUID Fn8000\\_001D\\_ECX\\_x0](#) reports topology information for the DC. See [CPUID Fn8000\\_001D\\_EAX\\_x0](#).

Bits	Description
31:0	<b>CacheNumSets: cache number of sets.</b> Cache number of sets is CacheNumSets+1. Value: 0000_003Fh.

### **CPUID Fn8000\_001D\_ECX\_x1 Cache Properties**

[CPUID Fn8000\\_001D\\_ECX\\_x1](#) reports topology information for the IC. See [CPUID Fn8000\\_001D\\_EAX\\_x0](#).

Bits	Description
31:0	<b>CacheNumSets: cache number of sets.</b> See: <a href="#">CPUID Fn8000_001D_ECX_x0</a> [CacheNumSets]. Value: 0000_00FFh.

### **CPUID Fn8000\_001D\_ECX\_x2 Cache Properties**

[CPUID Fn8000\\_001D\\_ECX\\_x2](#) reports topology information for the L2. See [CPUID Fn8000\\_001D\\_EAX\\_x0](#).

Bits	Description
31:0	<b>CacheNumSets: cache number of sets.</b> Value: Product-specific. See: <a href="#">CPUID Fn8000_001D_ECX_x0</a> [CacheNumSets].

### **CPUID Fn8000\_001D\_ECX\_x3 Cache Properties**

[CPUID Fn8000\\_001D\\_EAX\\_x3](#) reports done/null. See [CPUID Fn8000\\_001D\\_EAX\\_x0](#).

Bits	Description
31:0	Reserved. Value: 0000_0000h.

### CPUID Fn8000\_001D\_EDX\_x0 Cache Properties

CPUID Fn8000\_001D\_EDX\_x0 reports topology information for the DC. See [CPUID Fn8000\\_001D\\_EAX\\_x0](#).

Bits	Description
31:2	Reserved.
1	<b>CacheInclusive: cache inclusive.</b> Value: 0. 0=Cache is not inclusive of lower cache levels. 1=Cache is inclusive of lower cache levels.
0	<b>WBINVD: Write-Back Invalidate/Invalidate.</b> Value: 0. 0=WBINVD/INVD invalidates all lower level caches of non-originating cores sharing this cache. 1=WBINVD/INVD not ensured to invalidate all lower level caches of non-originating cores sharing this cache.

### CPUID Fn8000\_001D\_EDX\_x1 Cache Properties

CPUID Fn8000\_001D\_EDX\_x1 reports topology information for the IC. See [CPUID Fn8000\\_001D\\_EAX\\_x0](#).

Bits	Description
31:2	Reserved.
1	<b>CacheInclusive: cache inclusive.</b> Value: 0. See: <a href="#">CPUID Fn8000_001D_EDX_x0[CacheInclusive]</a> .
0	<b>WBINVD: Write-Back Invalidate/Invalidate.</b> Value: 0. See: <a href="#">CPUID Fn8000_001D_EDX_x0[WBINVD]</a> .

### CPUID Fn8000\_001D\_EDX\_x2 Cache Properties

CPUID Fn8000\_001D\_EDX\_x2 reports topology information for the L2. See [CPUID Fn8000\\_001D\\_EAX\\_x0](#).

Bits	Description
31:2	Reserved.
1	<b>CacheInclusive: cache inclusive.</b> See: <a href="#">CPUID Fn8000_001D_EDX_x0[CacheInclusive]</a> . Value: 1.
0	<b>WBINVD: Write-Back Invalidate/Invalidate.</b> Value: 0. See: <a href="#">CPUID Fn8000_001D_EDX_x0[WBINVD]</a> .

### CPUID Fn8000\_001D\_EDX\_x3 Cache Properties

CPUID Fn8000\_001D\_EAX\_x3 reports done/null. See [CPUID Fn8000\\_001D\\_EAX\\_x0](#).

Bits	Description
31:0	Reserved. Value: 0000_0000h.

**CPUID Fn8000\_001E\_EAX Extended APIC ID**

If **CPUID Fn8000\_0001\_ECX**[TopologyExtensions]==0 then **CPUID Fn8000\_001E\_E**[D,C,B,A]X is reserved. If (**MSR0000\_001B**[ApicEn]==0) then **CPUID Fn8000\_001E\_EAX**[ExtendedApicId] is reserved.

Bits	Description
31:0	<b>ExtendedApicId: extended APIC ID.</b> Value: IF ( <b>MSR0000_001B</b> [ApicEn]==0) THEN 0000_0000h ELSE <b>APIC20</b> [31:0] ENDIF.

**CPUID Fn8000\_001E\_EBX Compute Unit Identifiers**

See **CPUID Fn8000\_001E\_EAX**.

Bits	Description
31:16	Reserved.
15:8	<b>ThreadsPerComputeUnit: threads per compute unit.</b> The number of threads per compute unit is <b>ThreadsPerComputeUnit+1</b> . Value: 0.
7:0	<b>ComputeUnitId: compute unit ID.</b> Identifies the processor compute unit ID. Value: Product-specific.

**CPUID Fn8000\_001E\_ECX Node Identifiers**

See **CPUID Fn8000\_001E\_EAX**.

Bits	Description
31:0	Reserved.

**CPUID Fn8000\_001E\_EDX Reserved**

See **CPUID Fn8000\_001E\_EAX**.

Bits	Description
31:0	Reserved.

**3.18 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**

Bits	Description
63:0	Alias of <a href="#">MSR0000_0402</a> .

**MSR0000\_0001 Load-Store MCA Status**

Bits	Description
63:0	Alias of <a href="#">MSR0000_0401</a> .

**MSR0000\_0010 Time Stamp Counter (TSC)**

Reset: 0000\_0000\_0000\_0000h.

Bits	Description
63:32	<b>TSC[63:32]: time stamp counter high.</b> See: TSC[31:0].
31:0	<b>TSC[31:0]: time stamp counter low.</b> Read-write; updated-by-hardware. TSC[63:0] = {TSC[63:32],TSC[31:0]}. The TSC increments at the P0 frequency. This field uses software P-state numbering. See 2.5.3.1.1.1 [Software P-state Numbering]. The TSC counts at the same rate in all P-states, all C states, S0, or S1. A read of this MSR in guest mode is affected by <a href="#">MSRC000_0104 [Time Stamp Counter Ratio (TscRateMsr)]</a> . The value (TSC/TSCRatio) is the TSC P0 frequency based value (as if TSCRatio==1.0) when (TSCRatio!=1.0).

**MSR0000\_001B APIC Base Address (APIC\_BAR)**

Bits	Description
63:40	MBZ.
39:12	<b>ApicBar[39:12]: APIC base address register.</b> Read-write. Reset: 00_FEE0_0h. Specifies the base address, physical address [39:12], for the APICXX register set in xAPIC mode. See 2.4.8.1.2 [APIC Register Space].
11	<b>ApicEn: APIC enable.</b> Read-write. Reset: 0. See 2.4.8.1.2 [APIC Register Space]. 1=Local APIC is enabled in xAPIC mode.
10	MBZ.
9	MBZ.
8	<b>BSC: boot strap core.</b> Read-write; updated-by-hardware. Reset: x. 1=The core is the boot core of the BSP. 0=The core is not the boot core of the BSP.
7:0	MBZ.

**MSR0000\_002A Cluster ID (EBL\_CR\_POWERON)**

Read; GP-write. Writes to this register result in a GP fault with error code 0.

Bits	Description
63:18	MBZ.
17:16	<b>ClusterID</b> . Reset: 00b. The field does not affect hardware.
15:0	MBZ.

**MSR0000\_00E7 Max Performance Frequency Clock Count (MPERF)**

Reset: 0000\_0000\_0000\_0000h.

Bits	Description
63:0	<b>MPERF: maximum core clocks counter</b> . Read-write; <a href="#">Updated-by-hardware</a> . Incremented by hardware at the P0 frequency while the core is in C0. This register does not increment when the core is in the stop-grant state. In combination with <a href="#">MSR0000_00E8</a> , this is used to determine the effective frequency of the core. A read of this MSR in guest mode is affected by <a href="#">MSRC000_0104 [Time Stamp Counter Ratio (TscRateMsr)]</a> . This field uses software P-state numbering. See <a href="#">MSRC001_0015[Eff-FreqCntMwait]</a> , <a href="#">2.5.3.3 [Effective Frequency]</a> , and <a href="#">2.5.3.1.1.1 [Software P-state Numbering]</a> .

**MSR0000\_00E8 Actual Performance Frequency Clock Count (APERF)**

Reset: 0000\_0000\_0000\_0000h.

Bits	Description
63:0	<b>APERF: actual core clocks counter</b> . Read-write; <a href="#">Updated-by-hardware</a> . This register increments in proportion to the actual number of core clocks cycles while the core is in C0. The register does not increment when the core is in the stop-grant state. See <a href="#">MSR0000_00E7</a> .

**MSR0000\_00FE MTRR Capabilities (MTRRcap)**

Read; GP-write. Reset: 0000\_0000\_0000\_0508h.

Bits	Description
63:11	Reserved.
10	<b>MtrrCapWc: write-combining memory type</b> . 1=The write combining memory type is supported.
9	Reserved.
8	<b>MtrrCapFix: fixed range register</b> . 1=Fixed MTRRs are supported.
7:0	<b>MtrrCapVCnt: variable range registers count</b> . Specifies the number of variable MTRRs supported.

**MSR0000\_0174 SYSENTER CS (SYSENTER\_CS)**

Bits	Description
63:32	RAZ.



31:16	Reserved.
15:0	<b>SysEnterCS: SYSENTER target CS.</b> Read-write. Reset: 0000h. Holds the called procedure code segment.

**MSR0000\_0175 SYSENTER ESP (SYSENTER\_ESP)**

Reset: 0000\_0000\_0000\_0000h.

Bits	Description
63:32	Reserved.
31:0	<b>SysEnterESP: SYSENTER target SP.</b> Read-write. Holds the called procedure stack pointer.

**MSR0000\_0176 SYSENTER EIP (SYSENTER\_EIP)**

Reset: 0000\_0000\_0000\_0000h.

Bits	Description
63:32	Reserved.
31:0	<b>SysEnterEIP: SYSENTER target IP.</b> Read-write. Holds the called procedure instruction pointer.

**MSR0000\_0179 Global Machine Check Capabilities (MCG\_CAP)**

Read; GP-write.

Bits	Description
63:9	Reserved.
8	<b>McgCtlP: MCG_CTL register present.</b> Value: 1. 1=The machine check control registers (MCI_CTL) are present. See <a href="#">2.14.1 [Machine Check Architecture]</a>
7:0	<b>Count.</b> Value: 06h. Indicates the number of error reporting banks visible to each core. 06h=Error-reporting banks 0 through 5. See <a href="#">2.14.1.1 [Machine Check Registers]</a> .

**MSR0000\_017A Global Machine Check Status (MCG\_STAT)**Reset: 0000\_0000\_0000\_0000h. See [2.14.1 \[Machine Check Architecture\]](#).

Bits	Description
63:3	Reserved.
2	<b>MCIP: machine check in progress.</b> Read-write; set-by-hardware. 1=A machine check is in progress.
1	<b>EIPV: error instruction pointer valid.</b> 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	<b>RIPV: restart instruction pointer valid.</b> 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 (MCG\_CTL)**

Read-write. Reset: 0000\_0000\_0000\_0000h. This registers controls enablement of the individual error reporting banks; see 2.14.1 [Machine Check Architecture]. When a machine check register bank is not enabled in MCG\_CTL, errors for that bank are not logged or reported, and actions enabled through the MCA are not taken; each MCI\_CTL register identifies which errors are still corrected when MCG\_CTL[i] is disabled.

Bits	Description
63:7	Unused.
6	Unused.
5	<b>MC5En: MC5 register bank enable.</b> 1=The MC5 machine check register bank is enabled.
4	<b>MC4En: MC4 register bank enable.</b> 1=The MC4 machine check register bank is enabled for all cores of the node.
3	Unused.
2	<b>MC2En: MC2 register bank enable.</b> 1=The MC2 machine check register bank is enabled.
1	<b>MC1En: MC1 register bank enable.</b> 1=The MC1 machine check register bank is enabled.
0	<b>MC0En: MC0 register bank enable.</b> 1=The MC0 machine check register bank is enabled.

**MSR0000\_01D9 Debug Control (DBG\_CTL\_MSR)**

Bits	Description
63:7	Reserved.
6	MBZ.
5:2	<b>PB: performance monitor pin control.</b> Read-write. Reset: 0. This field does not control any hardware.
1	<b>BTF.</b> Read-write. Reset: 0. 1=Enable branch single step.
0	<b>LBR.</b> Read-write. Reset: 0. 1=Enable last branch record.

**MSR0000\_01DB Last Branch From IP (BR\_FROM)**

Read; GP-write; *Not-same-for-all*, *Updated-by-hardware*. Reset: 0000\_0000\_0000\_0000h.

Bits	Description
63:0	<b>LastBranchFromIP.</b> Loaded with the segment offset of the branch instruction.

**MSR0000\_01DC Last Branch To IP (BR\_TO)**

Read; GP-write; *Not-same-for-all*, *Updated-by-hardware*. Reset: 0000\_0000\_0000\_0000h.

Bits	Description
63:0	<b>LastBranchToIP.</b> Holds the target RIP of the last branch that occurred before an exception or interrupt.

**MSR0000\_01DD Last Exception From IP**Read; GP-write; **Not-same-for-all**, **Updated-by-hardware**. Reset: 0000\_0000\_0000\_0000h.

Bits	Description
63:0	<b>LastIntFromIP</b> . Holds the source RIP of the last branch that occurred before the exception or interrupt.

**MSR0000\_01DE Last Exception To IP**Read; GP-write; **Not-same-for-all**, **Updated-by-hardware**. Reset: 0000\_0000\_0000\_0000h.

Bits	Description
63:0	<b>LastIntToIP</b> . Holds the target RIP of the last branch that occurred before the exception or interrupt.

**MSR0000\_020[F:0] Variable-Size MTRRs Base/Mask**

Each MTRR ([MSR0000\\_020\[F:0\] \[Variable-Size MTRRs Base/Mask\]](#), [MSR0000\\_02\[6F:68,59:58,50\]](#), or [MSR0000\\_02FF \[MTRR Default Memory Type \(MTRRdefType\)\]](#)) specifies a physical address range and a corresponding memory type (MemType) associated with that range. Setting the memory type to an unsupported value results in a #GP.

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 [MSR0000\\_02FF\[MtrrDefTypeEn\]](#). A core access--with address CPUAddr--is determined to be within the address range of a variable-size MTRR if the following equation is true:

$$\text{CPUAddr}[39:12] \& \text{PhyMask}[39:12] == \text{PhyBase}[39:12] \& \text{PhyMask}[39:12].$$

For example, if the variable MTRR spans 256 KB and starts at the 1 MB address. The PhyBase would be set to 0\_0010\_0000h and the PhyMask to F\_FFFC\_0000h (with zeros filling in for bits[11:0]). This results in a range from 0\_0010\_0000h to 0\_0013\_FFFF.

**MSR0000\_020[E,C,A,8,6,4,2,0] Variable-Size MTRRs Base**Table 191: [Register Mapping](#) for [MSR0000\\_020\[E,C,A,8,6,4,2,0\]](#)

Register	Function
MSR0000_0200	Range 0
MSR0000_0202	Range 1
MSR0000_0204	Range 2
MSR0000_0206	Range 3
MSR0000_0208	Range 4
MSR0000_020A	Range 5
MSR0000_020C	Range 6
MSR0000_020E	Range 7

**Table 192: Valid Values** for Memory Type Definition

Bits	Description
000b	UC or uncacheable.
001b	WC or write combining.
011b-010b	Reserved
100b	WT or write through.
101b	WP or write protect.
110b	WB or write back.
111b	Reserved

Bits	Description
63:40	MBZ.
39:12	<b>PhyBase: base address.</b> Read-write. Reset: 0.
11:3	MBZ.
2:0	<b>MemType: memory type.</b> Read-write. Reset: 0. Address range from 00000h to 0FFFFh. See: <a href="#">Table 192 [Valid Values for Memory Type Definition]</a> .

**MSR0000\_020[F,D,B,9,7,5,3,1] Variable-Size MTRRs Mask**Table 193: [Register Mapping](#) for MSR0000\_020[F,D,B,9,7,5,3,1]

Register	Function
MSR0000_0201	Range 0
MSR0000_0203	Range 1
MSR0000_0205	Range 2
MSR0000_0207	Range 3
MSR0000_0209	Range 4
MSR0000_020B	Range 5
MSR0000_020D	Range 6
MSR0000_020F	Range 7

Bits	Description
63:40	MBZ.
39:12	<b>PhyMask: address mask.</b> Read-write. Reset: 0.
11	<b>Valid: valid.</b> Read-write. Reset: 0. 1=The variable-size MTRR pair is enabled.
10:0	MBZ.

**MSR0000\_02[6F:68,59:58,50] Fixed-Size MTRRs**

See [MSR0000\\_020\[F:0\]](#) for general MTRR information. Fixed MTRRs are enabled through [MSR0000\\_02FF\[MtrrDefTypeFixEn, MtrrDefTypeEn\]](#). For addresses below 1MB, the appropriate Fixed MTRRs override the default access destination. Each fixed MTRR includes two bits, RdDram and WrDram,

that determine the destination based on the access type.  
 See 2.4.6.1.2 [Determining The Access Destination for Core Accesses].

Table 194: Register Mapping for MSR0000\_02[6F:68,59:58,50]

Register	Function
MSR0000_0250	64K Range
MSR0000_0258	16K_0 Range
MSR0000_0259	16K_1 Range
MSR0000_0268	4K_0 Range
MSR0000_0269	4K_1 Range
MSR0000_026A	4K_2 Range
MSR0000_026B	4K_3 Range
MSR0000_026C	4K_4 Range
MSR0000_026D	4K_5 Range
MSR0000_026E	4K_6 Range
MSR0000_026F	4K_7 Range

Table 195: Field Mapping for MSR0000\_02[6F:68,59:58,50]

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	<b>RdDram: read DRAM.</b> See: MSR0000_02[6F:68,59:58,50][4].
59	<b>WrDram: write DRAM.</b> See: MSR0000_02[6F:68,59:58,50][3].
58:56	<b>MemType: memory type.</b> See: MSR0000_02[6F:68,59:58,50][2:0].
55:53	MBZ.
52	<b>RdDram: read DRAM.</b> See: MSR0000_02[6F:68,59:58,50][4].
51	<b>WrDram: write DRAM.</b> See: MSR0000_02[6F:68,59:58,50][3].
50:48	<b>MemType: memory type.</b> See: MSR0000_02[6F:68,59:58,50][2:0].
47:45	MBZ.

44	<b>RdDram: read DRAM.</b> See: <a href="#">MSR0000_02[6F:68,59:58,50][4]</a> .
43	<b>WrDram: write DRAM.</b> See: <a href="#">MSR0000_02[6F:68,59:58,50][3]</a> .
42:40	<b>MemType: memory type.</b> See: <a href="#">MSR0000_02[6F:68,59:58,50][2:0]</a> .
39:37	MBZ.
36	<b>RdDram: read DRAM.</b> See: <a href="#">MSR0000_02[6F:68,59:58,50][4]</a> .
35	<b>WrDram: write DRAM.</b> See: <a href="#">MSR0000_02[6F:68,59:58,50][3]</a> .
34:32	<b>MemType: memory type.</b> See: <a href="#">MSR0000_02[6F:68,59:58,50][2:0]</a> .
31:29	MBZ.
28	<b>RdDram: read DRAM.</b> See: <a href="#">MSR0000_02[6F:68,59:58,50][4]</a> .
27	<b>WrDram: write DRAM.</b> See: <a href="#">MSR0000_02[6F:68,59:58,50][3]</a> .
26:24	<b>MemType: memory type.</b> See: <a href="#">MSR0000_02[6F:68,59:58,50][2:0]</a> .
23:21	MBZ.
20	<b>RdDram: read DRAM.</b> See: <a href="#">MSR0000_02[6F:68,59:58,50][4]</a> .
19	<b>WrDram: write DRAM.</b> See: <a href="#">MSR0000_02[6F:68,59:58,50][3]</a> .
18:16	<b>MemType: memory type.</b> See: <a href="#">MSR0000_02[6F:68,59:58,50][2:0]</a> .
15:13	MBZ.
12	<b>RdDram: read DRAM.</b> See: <a href="#">MSR0000_02[6F:68,59:58,50][4]</a> .
11	<b>WrDram: write DRAM.</b> See: <a href="#">MSR0000_02[6F:68,59:58,50][3]</a> .
10:8	<b>MemType: memory type.</b> See: <a href="#">MSR0000_02[6F:68,59:58,50][2:0]</a> .
7:5	MBZ.
4	<b>RdDram: read DRAM.</b> IF ( <a href="#">MSRC001_0010[MtrrFixDramModEn]</a> ) THEN Read-write. ELSE MBZ. ENDIF. Reset: 0. 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM. Address range from 00000h to 0FFFFh. See: <a href="#">MSRC001_0010[MtrrFixDramEn, MtrrFixDramModEn]</a> .
3	<b>WrDram: write DRAM.</b> IF ( <a href="#">MSRC001_0010[MtrrFixDramModEn]</a> ) THEN Read-write. ELSE MBZ. ENDIF. Reset: 0. 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. Address range from 00000h to 0FFFFh. See: <a href="#">MSRC001_0010[MtrrFixDramEn, MtrrFixDramModEn]</a> .
2:0	<b>MemType: memory type.</b> Read-write. Reset: 0. Address range from 00000h to 0FFFFh. See: <a href="#">Table 192 [Valid Values for Memory Type Definition]</a> .

### MSR0000\_0277 Page Attribute Table (PAT)

This register specifies the memory type based on the PAT, PCD, and PWT bits in the virtual address page tables.

Bits	Description
63:59	MBZ.
58:56	<b>PA7MemType.</b> See: PA0MemType. Reset: 0h. Default UC. MemType for {PAT, PCD, PWT} = 7h.
55:51	MBZ.
50:48	<b>PA6MemType.</b> See: PA0MemType. Reset: 7h. Default UC-. MemType for {PAT, PCD, PWT} = 6h.
47:43	MBZ.

42:40	<b>PA5MemType.</b> See: PA0MemType. Reset: 4h. Default WT. MemType for {PAT, PCD, PWT} = 5h.																				
39:35	MBZ.																				
34:32	<b>PA4MemType.</b> See: PA0MemType. Reset: 6h. Default WB. MemType for {PAT, PCD, PWT} = 4h.																				
31:27	MBZ.																				
26:24	<b>PA3MemType.</b> See: PA0MemType. Reset: 0h. Default UC. MemType for {PAT, PCD, PWT} = 3h.																				
23:19	MBZ.																				
18:16	<b>PA2MemType.</b> See: PA0MemType. Reset: 7h. Default UC-. MemType for {PAT, PCD, PWT} = 2h.																				
15:11	MBZ.																				
10:8	<b>PA1MemType.</b> See: PA0MemType. Reset: 4h. Default WT. MemType for {PAT, PCD, PWT} = 1h.																				
7:3	MBZ.																				
2:0	<b>PA0MemType.</b> Read-write. Reset: 6h. MemType for {PAT, PCD, PWT} = 0h. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">Bits</th> <th style="text-align: left;">Description</th> <th style="text-align: left;">Bits</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>UC or uncacheable.4h</td> <td>WT or write through.</td> <td></td> </tr> <tr> <td>1h</td> <td>WC or write combining.5h</td> <td>WP or write protect.</td> <td></td> </tr> <tr> <td>2h</td> <td>MBZ.</td> <td>6h</td> <td>WB or write back.</td> </tr> <tr> <td>3h</td> <td>MBZ.</td> <td>7h</td> <td>UC- or uncacheable (overridden by WC state).</td> </tr> </tbody> </table>	Bits	Description	Bits	Description	0h	UC or uncacheable.4h	WT or write through.		1h	WC or write combining.5h	WP or write protect.		2h	MBZ.	6h	WB or write back.	3h	MBZ.	7h	UC- or uncacheable (overridden by WC state).
Bits	Description	Bits	Description																		
0h	UC or uncacheable.4h	WT or write through.																			
1h	WC or write combining.5h	WP or write protect.																			
2h	MBZ.	6h	WB or write back.																		
3h	MBZ.	7h	UC- or uncacheable (overridden by WC state).																		

**MSR0000\_02FF MTRR Default Memory Type (MTRRdefType)**

See [MSR0000\\_020\[F:0\]](#) for general MTRR information.

Bits	Description
63:12	MBZ.
11	<b>MtrrDefTypeEn: variable and fixed MTRR enable.</b> Read-write. Reset: 0. 1= <a href="#">MSR0000_020[F:0]</a> [Variable-Size MTRRs Base/Mask], and <a href="#">MSR0000_02[6F:68,59:58,50]</a> [Fixed-Size MTRRs] are enabled. 0=Fixed and variable MTRRs are not enabled.
10	<b>MtrrDefTypeFixEn: fixed MTRR enable.</b> Read-write. Reset: 0. 1= <a href="#">MSR0000_02[6F:68,59:58,50]</a> [Fixed-Size MTRRs] are enabled. This field is ignored (and the fixed MTRRs are not enabled) if <a href="#">MSR0000_02FF[MtrrDefTypeEn]=0</a> .
9:8	MBZ.
7:0	<b>MemType: memory type.</b> Read-write. Reset: 0. If MtrrDefTypeEn==1 then MemType specifies the memory type for memory space that is not specified by either the fixed or variable range MTRRs. If MtrrDefTypeEn==0 then the default memory type for all of memory is UC. Valid encodings are {00000b, <a href="#">MSR0000_02[6F:68,59:58,50][2:0]</a> }.

**MSR0000\_0400 MC0 Machine Check Control (MC0\_CTL)**

Read-write. Reset: 0000\_0000\_0000\_0000h. See 2.14.1 [Machine Check Architecture]. See [MSRC001\\_0044](#) [DC Machine Check Control Mask (MC0\_CTL\_MASK)]

Bits	Description
63:12	Unused.
11	<b>SRDE_ALL: all system read data errors.</b> Report system read data errors for any operation including a DC/IC fetch, or TLB reload.

10	<b>SRDET: read data errors on TLB reload.</b> Report system read data errors on a TLB reload if <a href="#">MSR0000_0400[SRDE_ALL]</a> = 1.
9	<b>SRDES: read data errors on store.</b> Report system read data errors on a store if <a href="#">MSR0000_0400[SRDE_ALL]</a> = 1.
8	<b>SRDEL: read data errors on load.</b> Report system read data errors on a load if <a href="#">MSR0000_0400[SRDE_ALL]</a> = 1.
7	<b>TLB2M: L2DTLB multi-hit errors.</b> Report L2DTLB multi-hit errors.
6	<b>TLB2P: L2DTLB parity errors.</b> Report TLB parity and multimatch errors.
5	<b>TLB1M: L1DTLB multi-hit errors.</b> Report L1DTLB multi-hit errors.
4	Unused.
3	<b>DTP: tag array parity errors.</b> Report data cache tag array parity errors.
2	<b>DDP: data array parity errors.</b> Report data cache data array parity errors.
1:0	Unused.

### **MSR0000\_0401 MC0 Machine Check Status (MC0\_STATUS)**

See 2.14.1 [Machine Check Architecture]. See [MSRC001\\_0015\[McStatusWrEn\]](#). Table 197 describes the error codes and status register settings for each error type. [MSR0000\\_0001](#) is an alias of [MSR0000\\_0401](#).

Bits	Description
63	<b>Val: error valid.</b> Read-write; set-by-hardware. Cold reset: 0. 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.
62	<p><b>Overflow: error overflow.</b> Read-write; set-by-hardware. Cold reset: 0. 1=An error was detected while the valid bit (Val) was set; at least one error was not logged. Overflow is set independently of whether the existing error is overwritten.</p> <p>The following hierarchy identifies the error logging priorities.</p> <ol style="list-style-type: none"> <li>1. Uncorrectable errors</li> <li>2. Correctable errors</li> </ol> <p>The machine check mechanism handles the contents of <code>MCi_STATUS</code> during overflow as follows:</p> <ul style="list-style-type: none"> <li>• Higher priority errors overwrite lower priority errors.</li> <li>• New errors of equal or lower priority do not overwrite existing errors.</li> <li>• Uncorrectable errors which are not logged due to overflow result in setting PCC, unless the new uncorrectable error is of the same type and in the same reportable address range as the existing error.</li> </ul>
61	<b>UC: error uncorrected.</b> Read-write; Updated-by-hardware. Cold reset: 0. 1=The error was not corrected by hardware.
60	<b>En: error enable.</b> Read-write; Updated-by-hardware. Cold reset: 0. 1=MCA error reporting is enabled for this error, as indicated by <code>MCi_CTL</code> .
59	<p><b>MiscV: miscellaneous error register valid.</b></p> <p>Value: 0. 1=Valid thresholding in <a href="#">MSR0000_0403</a>.</p>



58	<b>AddrV: error address valid.</b> Read-write; Updated-by-hardware. Cold reset: 0. 1=MCi_ADDR contains address information associated with the error.																				
57	<b>PCC: processor context corrupt.</b> Read-write; Updated-by-hardware. Cold reset: 0. 1=Hardware context held by the processor may have been corrupted. Continued operation of the system may have unpredictable results. The error is not recoverable or survivable, and the system should be reinitialized. See <a href="#">2.14.1.6.1 [Differentiation Between System-Fatal and Process-Fatal Errors]</a> .																				
56:45	Reserved.																				
44	<b>Deferred: deferred error.</b> Value: 0. 1=A deferred error was created. A deferred error is the result of an uncorrectable data error which did not immediately cause a processor exception; the data is poisoned and an exception is deferred until the data is loaded by a core.																				
43	<b>Poison: poison error.</b> Value: 0. 1=The error was the result of attempting to consume poisoned data. This indicator does not apply to <a href="#">MSR0000_0411 [MC4 Machine Check Status (MC4_STATUS)]</a> .																				
42:40	Reserved.																				
39	Reserved.																				
38:36	<b>Way: cache way in error.</b> Read-write; Updated-by-hardware.Cold reset: 0. Indicates the cache way for parity error only. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th><u>Bits</u></th> <th><u>Description</u></th> <th><u>Bits</u></th> <th><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Way 0</td> <td>4h</td> <td>Way 4</td> </tr> <tr> <td>1h</td> <td>Way 1</td> <td>5h</td> <td>Way 5</td> </tr> <tr> <td>2h</td> <td>Way 2</td> <td>6h</td> <td>Way 6</td> </tr> <tr> <td>3h</td> <td>Way 3</td> <td>7h</td> <td>Way 7</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Description</u>	<u>Bits</u>	<u>Description</u>	0h	Way 0	4h	Way 4	1h	Way 1	5h	Way 5	2h	Way 2	6h	Way 6	3h	Way 3	7h	Way 7
<u>Bits</u>	<u>Description</u>	<u>Bits</u>	<u>Description</u>																		
0h	Way 0	4h	Way 4																		
1h	Way 1	5h	Way 5																		
2h	Way 2	6h	Way 6																		
3h	Way 3	7h	Way 7																		
35:21	Reserved.																				
20:16	<b>ErrorCodeExt: extended error code.</b> Read-write; Updated-by-hardware. Cold reset: 0. Logs an extended error code when an error is detected. This model-specific field is used in conjunction with Error Code to identify the error sub-type for root cause analysis (see <a href="#">2.14.1.5 [Error Code]</a> ). See <a href="#">Table 197</a> for expected values.																				
15:0	<b>ErrorCode: error code.</b> Read-write; Updated-by-hardware. Cold reset: 0. See <a href="#">2.14.1.5 [Error Code]</a> for details on decoding this field. See <a href="#">Table 197</a> for expected values.																				

**Table 196: MC0 Error Descriptions**

Error Type	Error Sub-type	Description <sup>1</sup>	CTL <sup>2</sup>	EAC <sup>3</sup>
Data Parity		A DC data parity error for a tag hit occurred during an LS access to the DC.	DDP	E
Tag Parity		A DC tag array parity error for a data hit occurred during an LS access to the DC.	DTP	E
Copyback Parity		A DC data or tag array parity error for a tag miss occurred during an LS access to the DC.	DDP, DTP	E

**Table 196: MC0 Error Descriptions**

Error Type	Error Sub-type	Description <sup>1</sup>	CTL <sup>2</sup>	EAC <sup>3</sup>
Tag Snoop Parity		A tag parity error was encountered during snoop. Data array parity check enabled only for tag hit.	DDP, DTP	E
L2 TLB Error	TLB parity	Parity error in BTLB.  Regardless of whether this error is masked from logging, it will invalidate the lookup index (for both 4k and 2M storage) where the error occurred and an L2DTLB miss will be generated.	TLB2P	E
L1 TLB Error	TLB multimatch	Hit multiple entries.  Regardless of whether this error is masked from logging or enabled for SyncFlood generation, the L1DTLB will be flushed.	TLB1M	E
L2 TLB Error	TLB multimatch	Hit multiple entries.  Regardless of whether this error is masked from logging, it will invalidate the lookup index (for both 4k and 2M storage) where the error occurred and an L2DTLB miss will be generated.	TLB2M	E
System Read Data Error	TLB Reload	System read data error occurred on a TLB reload.	SRDET	E
	Store	System read data error occurred on a store.	SRDES	E
	Load	System read data error occurred on a load .	SRDEL	E

1. CID: core ID. All LS errors are reported to the affected core; see 2.14.1.3 [Error Detection, Action, Logging, and Reporting].  
 2. See MSR0000\_0400.  
 3. EAC: D=Error action taken if detected. E=Error action taken if MCA bank enabled. See 2.14.1.3 [Error Detection, Action, Logging, and Reporting].

**Table 197: MC0 Error Signatures**

Error Type	Error	ErrorCodeExt	Error Code						UC	ADDRV	PCC
			Type	UU/PP	T	RRRR	II/TT	LL			
Data Parity	Load,	0h	MEM	-	-	DRD	Data	L1	1	1	0
	Store	0h		-	-	DWR	Data	L1	1	1	1
Tag Parity	Load,	10h		-	-	DRD	Data	L1	1	1	1
	Store	10h		-	-	DWR	Data	L1	1	1	1
Copyback Parity	-	0h		-	-	Evict	Data	L1	1	1	1
Tag Snoop Parity	-	10h		-	-	Snoop	Data	L1	1	1	1
L2DTLB Parity	-	0h	TLB	-	-	-	Data	L2	0	1	0
L1DTLB Multimatch	-	1h		-	-	-	Data	L1	0 <sup>1</sup>	1	0 <sup>1</sup>
L2DTLB Multimatch	-	1h		-	-	-	Data	L2	0	1	0

**Table 197: MC0 Error Signatures**

Error Type	Error	ErrorCodeExt	Error Code						UC	ADDR	PCC
			Type	UU/PP	T	RRRR	II/TT	LL			
Read Data Error on TLB Reload	-	0h	BUS	SRC	0	RD	MEM/IO	LG	1	1	1
Read Data Error on Store	-	0h		SRC	0	DWR	MEM/IO	LG	1	1	1
Read Data Error on Load	-	0h		SRC	0	DRD	MEM/IO	LG	1	1	1
1. If an MCA error causes an unlocking store to miss in both the L1DTLB and L2DTLB, the error is uncorrectable. Both PCC and UC will be set to '1'.											

**MSR0000\_0402 MC0 Machine Check Address (MC0\_ADDR)**

Read-write; Updated-by-hardware. Cold reset: 0000\_0000\_0000\_0000h. The MCi\_ADDR register contains valid data if indicated by MCi\_STATUS[AddrV]. See 2.14.1 [Machine Check Architecture]. MSR0000\_0000 is an alias of MSR0000\_0402.

Bits	Description
63:0	<b>ADDR: Address.</b> See Table 198.

**Table 198: MC0 Address Register**

Error Type	Error Sub-type	Bits	Description
Data or Tag Parity	Load, Store, HW Prefetch	39:4	<b>PhysAddr[39:4].</b>
Copyback Parity		11:6	<b>PhysAddr[11:6].</b>
Tag Snoop Parity		39:6	<b>PhysAddr[39:6].</b>
BTLB Parity		47:12	<b>LinAddr[47:12].</b>
BTLB Multimatch			
Read Data Error	TLB Reload, Load, or Store	39:4	<b>PhysAddr[39:4].</b>

**MSR0000\_0403 MC0 Machine Check Miscellaneous (MC0\_MISC)**

Bits	Description
63:0	Reserved.

**MSR0000\_0404 MC1 Machine Check Control (MC1\_CTL)**

Read-write.

Reset: 0000\_0000\_0000\_0000h. See 2.14.1 [Machine Check Architecture].

Bits	Description
63:13	Unused.
12	Unused.
11:10	Unused.
9	<b>SRDE: read data errors.</b> Report system read data errors for an instruction cache fetch if <a href="#">MSR0000_0400[SRDE_ALL]</a> = 1.
8:7	Unused.
6	<b>TLBP: TLB parity errors.</b> Report TLB parity errors.BTLB Parity and Multimatch.
5	Unused.
4	<b>ISTP: snoop tag array parity errors.</b> Report instruction cache snoop tag array parity errors which occur during snoop.
3	<b>ITP: tag array parity errors.</b> Report instruction cache tag array parity errors.
2	<b>IDP: data array parity errors.</b> Report instruction cache data array parity errors.
1	<b>IVP: victim parity errors.</b> Report victim parity errors.
0	<b>PRPE: ME patch RAM parity error.</b> Report patch ram parity errors.

**MSR0000\_0405 MC1 Machine Check Status (MC1\_STATUS)**

See 2.14.1 [Machine Check Architecture]. See [MSRC001\\_0015](#)[McStatusWrEn]. [Table 199](#) describes each error type. [Table 200](#) describes the error codes and status register settings for each error type.

Bits	Description								
63	<b>Val: error valid.</b> See: <a href="#">MSR0000_0401</a> [Val].								
62	<b>Overflow: error overflow.</b> See: <a href="#">MSR0000_0401</a> [Overflow].								
61	<b>UC: error uncorrected.</b> See: <a href="#">MSR0000_0401</a> [UC].								
60	<b>En: error enable.</b> See: <a href="#">MSR0000_0401</a> [En].								
59	<b>MiscV: miscellaneous error register valid.</b> Value: 0. See: <a href="#">MSR0000_0401</a> [MiscV]. 1=Valid thresholding in <a href="#">MSR0000_0407</a> .								
58	<b>AddrV: error address valid.</b> See: <a href="#">MSR0000_0401</a> [AddrV].								
57	<b>PCC: processor context corrupt.</b> See: <a href="#">MSR0000_0401</a> [PCC].								
56:40	Reserved.								
39:36	<b>Way: cache way in error.</b> Read-write; Updated-by-hardware.Cold reset: 0. Indicates the cache way in error. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Way 0</td> </tr> <tr> <td>1h</td> <td>Way 1</td> </tr> <tr> <td>Fh-2h</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Description	0h	Way 0	1h	Way 1	Fh-2h	Reserved
Bits	Description								
0h	Way 0								
1h	Way 1								
Fh-2h	Reserved								

35:28	Reserved.
27:24	<b>PRParity[3:0]: ME Patch Ram Parity [3:0]</b> . Read-write; updated-by-hardware. Cold reset: 0.
23:21	Reserved.
20:16	<b>ErrorCodeExt: extended error code</b> . Read-write; updated-by-hardware. Cold reset: 0. See <a href="#">Table 200</a> for expected values.
15:0	<b>ErrorCode: error code</b> . Read-write; updated-by-hardware. Cold reset: 0. See <a href="#">2.14.1.5 [Error Code]</a> for details on decoding this field. See <a href="#">Table 200</a> for expected values.

**Table 199: MC1 Error Descriptions**

Error Type	Error	Description	CTL <sup>3</sup>	CID <sup>2</sup>	EAC <sup>1</sup>
Read Data Error		An error occurred during an attempted read of data from the NB. Possible reasons include master abort and target abort.	SRDE		
Data or Tag Parity		An IC data or tag array parity error for a tag hit occurred during instruction fetch from the IC. The data is discarded from the IC, the data array and all tag arrays are cleared, and the line is refetched.	IDP, ITP		
Tag Snoop		A tag error was encountered during snoop or victimization.	ISTP		
Victim Parity		IC Tag parity error as a result of victim cast-out.	IVP		
BTLB Parity		Parity error in BTLB.	TLBP		
BTLB Multimatch		Hit multiple entries.	TLBP		
PRPE PatchRam Parity		Microcode Patch RAM Parity Error	PRPE		

1. EAC: D=Error action taken if detected. E=Error action taken if MCA bank enabled. See [2.14.1.3 \[Error Detection, Action, Logging, and Reporting\]](#).

2. CID: core ID. A=Error reported to the affected core. 0=Error reported to core 0 of the L2 complex. B=Error reported to all cores of the L2 complex. See [2.14.1.3 \[Error Detection, Action, Logging, and Reporting\]](#).

3. See [MSR0000\\_0404](#).

**Table 200: MC1 Error Signatures**

Error Type	Error	ErrorCod eExt	Error Code						UC	ADDRV	PCC
			Type	PP	TT	RRRR	II/TT	LL			
Read Data Error		0h	BUS	SRC	0	IRD	MEM	LG	1	1	0
Data Parity		0h	MEM	-	-	IRD	Instr	L1	0	1	0
Tag Parity		1h	MEM	-	-	IRD	Instr	L1	0	1	0
Tag Snoop		0h	MEM	-	-	Snoop	Instr	L1	1	1	1
Victim Parity		0h	MEM	-	-	-	Instr	L1	0	1	0

**Table 200: MC1 Error Signatures**

Error Type	Error	ErrorCodeExt	Error Code						UC	ADDRV	PCC
			Type	PP	TT	RRRR	II/TT	LL			
BTLB Parity		0h	TLB	-	-	-	Instr	L2	0	1	0
BTLB Multimatch		1h	TLB	-	-	-	Instr	L2	0	1	0
PRPE PatchRam Parity		2h	MEM	-	-	-	Instr	LG	1	1	0

**MSR0000\_0406 MC1 Machine Check Address (MC1\_ADDR)**

Read-write; Updated-by-hardware. Cold reset: 0000\_0000\_0000\_0000h. The MC<sub>i</sub>\_ADDR register contains valid data if indicated by MC<sub>i</sub>\_STATUS[AddrV]. See 2.14.1 [Machine Check Architecture].

Bits	Description
63:0	<b>ADDR:</b> Address. See Table 201.

The following table defines the address register as a function of error type.

**Table 201: MC1 Address Register**

Error Type	Error Sub-Type	Bits	Description
IC Data and Tag Parity		47:4	Linear address
Tag Snoop		39:6	Physical address
Victim Parity		47:6	Linear Address
BTLB Parity		47:4	Linear address
BTLB Multimatch			
PRPE PatchRam ParityError		12:0	Patch Ram Address
BU Read Data Error	-	63:40	Reserved.
		39:6	<b>PhysAddr[47:6].</b>
		5:0	Reserved.

**MSR0000\_0407 MC1 Machine Check Miscellaneous (MC1\_MISC)**

Bits	Description
63:0	Reserved.

**MSR0000\_0408 MC2 Machine Check Control (MC2\_CTL)**

Read-write; Per-L2; not-same-for-all.

Reset: 0000\_0000\_0000\_0000h. See 2.14.1 [Machine Check Architecture]. See MSRC001\_0046 [BU Machine Check Control Mask (MC2\_CTL\_MASK)].

Bits	Description
63:14	Unused.
13	<b>AttrParity: attribute parity errors.</b>
12:9	Unused.
8	<b>DataUncor: uncorrectable data errors.</b>
7	<b>DataCor: correctable data errors.</b>
6	<b>DataParity: data parity errors.</b>
5	<b>TagUncor: uncorrectable tag errors.</b>
4	<b>TagCor: correctable tag errors.</b>
3	Unused.
2	<b>IBUFF: NB read response error due to a fill.</b>
1	<b>OBUFF: NB read response error due to a write/victim.</b>
0	Unused.

#### **MSR0000\_0409 MC2 Machine Check Status (MC2\_STATUS)**

IF (MSRC001\_10A0[McaToMstCoreEn]) THEN Per-L2; not-same-for-all. ENDIF. See 2.14.1 [Machine Check Architecture]. See MSRC001\_0015[McStatusWrEn]. Table 202 describes each error type. Table 203 describes the error codes and status register settings for each error type. See MSRC001\_10A0[McaToMstCoreEn].

Bits	Description
63	<b>Val: error valid.</b> See: MSR0000_0401[Val].
62	<b>Overflow: error overflow.</b> See: MSR0000_0401[Overflow].
61	<b>UC: error uncorrected.</b> See: MSR0000_0401[UC].
60	<b>En: error enable.</b> See: MSR0000_0401[En].
59	<b>MiscV: miscellaneous error register valid.</b> Value: 0. See: MSR0000_0401[MiscV]. 1=Valid thresholding in MSR0000_040B.
58	<b>AddrV: error address valid.</b> See: MSR0000_0401[AddrV].
57	<b>PCC: processor context corrupt.</b> See: MSR0000_0401[PCC].
56	<b>ErrCoreIdVal.</b> Read-write; Updated-by-hardware. Cold reset: 0. 1=The ErrCoreId field is valid.
55	Reserved.

54:47	<p><b>Syndrome[7:0]</b>. Read-write; Updated-by-hardware. Cold reset: 0. The syndrome bits when an ECC error is detected. See <a href="#">Table 203</a> for when Syndrome is valid.                  Syndrome[15:0] = {Syndrome[15:8], Syndrome[7:0]}.</p> <table border="1"> <thead> <tr> <th>Array</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>L2 Tag</td> <td>Syndrome[7:0]. (Syndrome[15:8] is 0)</td> </tr> <tr> <td>L2 Data</td> <td>Syndrome[8:0]. (Syndrome[15:9] is 0)</td> </tr> <tr> <td>NbRead, NbFill</td> <td>Syndrome[15:0]. Even parity.</td> </tr> </tbody> </table>	Array	Description	L2 Tag	Syndrome[7:0]. (Syndrome[15:8] is 0)	L2 Data	Syndrome[8:0]. (Syndrome[15:9] is 0)	NbRead, NbFill	Syndrome[15:0]. Even parity.		
Array	Description										
L2 Tag	Syndrome[7:0]. (Syndrome[15:8] is 0)										
L2 Data	Syndrome[8:0]. (Syndrome[15:9] is 0)										
NbRead, NbFill	Syndrome[15:0]. Even parity.										
46	<b>CECC: correctable ECC error</b> . Read-write; Updated-by-hardware. Cold reset: 0. 1=The error was a correctable ECC error.										
45	<b>UECC: uncorrectable ECC error</b> . Read-write; updated-by-hardware. Cold reset: 0. 1=The error was an uncorrectable ECC error.										
44	<b>Deferred: deferred error</b> . See: <a href="#">MSR0000_0401</a> [Deferred].										
43	<b>Poison: poison error</b> . See: <a href="#">MSR0000_0401</a> [Poison].										
42	Reserved.										
41:36	<p><b>Way: cache way/entry in error</b>. Read-write; Updated-by-hardware. Cold reset: 0. Indicates the cache way in error. See <a href="#">Table 203</a> for when Way is valid and what ways are valid.</p> <pre>                 IF ((ErrorCodeExt==00100b)) THEN                     CRQ IBUFF error location specified by {Way, <a href="#">MSR0000_040A</a>[5:4]}.                 ELSEIF ((ErrorCodeExt==00101b)) THEN                     CRQ OBUFF error location specified by {Way, <a href="#">MSR0000_040A</a>[5:4]}.                 ELSE                     <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00h</td> <td>Way 0</td> </tr> <tr> <td>01h</td> <td>Way 1</td> </tr> <tr> <td>0Eh-02h</td> <td>Way &lt;Way&gt;</td> </tr> <tr> <td>0Fh</td> <td>Way 15</td> </tr> </tbody> </table>                 ENDIF             </pre>	Bits	Description	00h	Way 0	01h	Way 1	0Eh-02h	Way <Way>	0Fh	Way 15
Bits	Description										
00h	Way 0										
01h	Way 1										
0Eh-02h	Way <Way>										
0Fh	Way 15										
35:32	<b>ErrCoreId</b> . Read-write; Updated-by-hardware. Cold reset: 0. When ErrCoreIdVal=1, this field indicates which core within the processor is associated with the error; Otherwise this field is reserved.										
31:24	<b>Syndrome[15:8]</b> . See: <a href="#">MSR0000_0409</a> [Syndrome[7:0]].										
23:21	Reserved.										
20:16	<b>ErrorCodeExt: extended error code</b> . Read-write; updated-by-hardware. Cold reset: 0. See <a href="#">MSR0000_0401</a> [ErrorCodeExt]. See <a href="#">Table 203</a> for expected values.										
15:0	<b>ErrorCode: error code</b> . Read-write; Updated-by-hardware. Cold reset: 0. See <a href="#">2.14.1.5 [Error Code]</a> for details on decoding this field. See <a href="#">Table 203</a> for expected values.										

**Table 202: MC2 Error Descriptions**

Error Type	Error	Description	CTL <sup>3</sup>	CID <sup>2</sup>	EAC <sup>1</sup>
NB	IBUFF	A parity error occurred in the IBUFF.	IBUFF	-	E <sup>1</sup>
NB	OBUFF	A parity error occurred in the OBUFF.	OBUFF	-	E <sup>1</sup>



**Table 202: MC2 Error Descriptions**

Error Type	Error	Description	CTL <sup>3</sup>	CID <sup>2</sup>	EAC <sup>1</sup>
L2 Cache	L2Tag	A correctable or uncorrectable ECC error was seen in the L2 tag.	TagCor, TagUncor	-	E <sup>1</sup>
L2 Cache	L2Data	A correctable or uncorrectable ECC error was seen in the L2 data array.	DataParity, DataCor, DataUncor.	-	E <sup>1</sup>
Attribute Parity	Par	A parity error was seen in the L2 attribute bits. This error was corrected by hardware.	AttrParity	A <sup>2</sup>	E <sup>1</sup>

1. EAC: The error action is taken if detected for all CU errors. D=Error action taken if detected. E=Error action taken if MCA bank enabled. See 2.14.1.3 [Error Detection, Action, Logging, and Reporting]. If any core in the L2 complex has MCG\_CTL[2] asserted, the MCA logic in the L2I is enabled.
2. CID: core ID. A=Error reported to the affected core. 0=Error reported to core 0 of the L2 complex; see 2.14.1.3 [Error Detection, Action, Logging, and Reporting].
3. See MSR0000\_0408.
4. Error Action: Sync flood=Take sync flood if PCC=1. None=No action other than that specified by MCA.

**Table 203: MC2 Error Signatures**

Error Type	Error Sub-Type	Sub	Error Code Ext	Error Code						UC	ADDRV	PCC	Syndrome	Way	CECC	UECC	ErrCoreIdVal
				Type	PP	T	RRRR	II/TT	LL								
NB	IBUFF	-	04h	MEM	-	-	RD	G	L2	1	1	1	15:0	5:0	0	0	0
	OBUFF		05h				WR										
Tag	Cor	BankReq	09h	MEM	-	-	GEN	G	L2	0	1	0	7:0	3:0	1/0	0	1
		Prb	0Ah				Probe										0
		Fill	0Bh				RD										1
Tag	Uncor	BankReq	0Dh	MEM	-	-	GEN	G	L2	1	1	1	7:0	3:0	0	1	1
		Prb	0Eh				Probe										0
		Fill	0Fh				RD										1
Data	Cor	Hit	10h	MEM	-	-	RD	G	L2	0	1	0	8:0	3:0	1/0	0	1
		Attr	11h				GEN										1
		Vict	12h				Evict										0
		Fill	13h				RD										1
Data	Uncor	Hit	14h	MEM	-	-	RD	G	L2	1	1	1	8:0	3:0	0	1	1
		Attr	15h				GEN										1
		Vict	16h				Evict										0
		Fill	17h				RD										1
Data	Par	Hit	18h	MEM	-	-	RD	G	L2	0	1	0	-	3:0	0	0	1
		Attr	19h				GEN										
		Fill	1Bh				IRD										
Attr	Par	Hit	1Ch	MEM	-	-	RD	G	L2	0	1	0	-	3:0	0	0	1
		Attr	1Dh				GEN										
		Fill	1Fh				IRD										

**MSR0000\_040A MC2 Machine Check Address (MC2\_ADDR)**

Read-write; Updated-by-hardware. Cold reset: 0000\_0000\_0000\_0000h. See 2.14.1 [Machine Check Architecture]. The following table defines the address register as a function of error type.

This register is shared by all cores of the L2 complex when the `MSRC001_10A0[McaToMstCoreEn]==0`.

When (`MSRC001_10A0[McaToMstCoreEn]==1`), only the lowest enabled core in the complex can read/write this register. All other cores will read all 0s and writes will be ignored.

Bits	Description
63:40	Reserved.
39:4	<b>ADDR</b> . Read-write; updated-by-hardware. See Table 204.
3:0	Reserved.

**Table 204: MC2 Address Register**

Error Type	Error Sub-Type	Bits	Description
L2 Tag	ErrorCodeExt=0 9h to 0Fh.	63:17	Reserved.
		16:6	<b>PhysAddr[16:6]</b> .
		5:0	Reserved.
Data, Attr, NB	ErrorCodeExt=0 4h to 05h, 10h to 1Fh.	63:40	RAZ
		39:4	<b>PhysAddr[39:4]</b> .
		3:0	RAZ

**MSR0000\_040B MC2 Machine Check Miscellaneous (MC2\_MISC)**

Bits	Description
63:0	Reserved.

**MSR0000\_040C MC3 Machine Check Control (MC3\_CTL)**

Reset: 0000\_0000\_0000\_0000h. Read-only.

Bits	Description
63:0	Unused.

**MSR0000\_040D MC3 Machine Check Status (MC3\_STATUS)**

Reset: 0. See `MSRC001_0015[McStatusWrEn]`.

Bits	Description
63:0	Reserved.

**MSR0000\_040E MC3 Machine Check Address (MC3\_ADDR)**

Reset: 0000\_0000\_0000\_0000h. Read-only.

Bits	Description
63:0	Reserved.

**MSR0000\_040F MC3 Machine Check Miscellaneous (MC3\_MISC)**

Reset: 0000\_0000\_0000\_0000h. Read-only.

Bits	Description
63:0	Reserved.

**MSR0000\_0410 MC4 Machine Check Control (MC4\_CTL)**

Read-write; **Not-same-for-all**. Reset: 0000\_0000\_0000\_0000h.

[MSR0000\\_0410](#)[31:0] is an alias of [D18F3x40](#), which is accessible through PCI configuration space. Only one of these registers exists in multi-core devices; see [3.1.1 \[Northbridge MSRs In Multi-Core Products\]](#).

Accessibility of this register by non-NBC cores is affected by [D18F3x44](#)[NbMcaToMstCpuEn].

See [D18F3x44 \[MCA NB Configuration\]](#) for further NB MCA configuration controls. See [2.14.1 \[Machine Check Architecture\]](#) for a general description of the machine check architecture. See [MSRC001\\_0048 \[NB Machine Check Control Mask \(MC4\\_CTL\\_MASK\)\]](#) for the corresponding error mask register.

Bits	Description
63:32	Unused.
31	<b>McaCpuDatErrEn: L2 complex data error.</b> 1=Enables MCA reporting of CPU data errors sent to the NB.
30	Unused.
29:28	Unused.
27	Unused.
26	<b>NbArrayParEn: northbridge array parity error reporting enable.</b> 1=Enables reporting of parity errors in the NB arrays.
25	<b>UsPwDatErrEn: upstream data error enable.</b> Read-write. 1=Enables MCA reporting of upstream posted writes in which the EP bit is set.
24:18	Unused.
17	<b>CpPktDatEn: completion packet error reporting enable.</b> Read-write. 1=Enables MCA reporting of completion packets with the EP bit set.
16	<b>NbIntProtEn: northbridge internal bus protocol error reporting enable.</b> Read-write. 1=Enables MCA reporting of protocol errors detected on the northbridge internal bus. When possible, this enable should be cleared before initiating a warm reset to avoid logging spurious errors due to RESET_L signal skew.
15:13	Unused.
12	<b>WDTRptEn: watchdog timer error reporting enable.</b> 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 <a href="#">D18F3x44 [MCA NB Configuration]</a> 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	<b>AtomicRMWEn: atomic read-modify-write error reporting enable.</b> 1=Enables MCA reporting of atomic read-modify-write (RMW) commands received from an IO 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	Unused.
9	<b>TgtAbortEn: target abort error reporting enable.</b> 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	<b>MstrAbortEn: master abort error reporting enable.</b> 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	Unused.
5	<b>SyncPktEn: link sync packet error reporting enable.</b> 1=Enables MCA reporting of link-defined sync error packets detected on link. The NB floods its outgoing link with sync packets after detecting a sync packet on the incoming link independent of the state of this bit.
4:2	Unused.
1	<b>UECCEn: uncorrectable ECC error reporting enable.</b> 1=Enables MCA reporting of DDR3 DRAM uncorrectable ECC errors which are detected in the NB. In some cases data may be forwarded to the core prior to checking ECC in which case the check takes place in one of the other error reporting banks.
0	<b>CECCEn: correctable ECC error reporting enable.</b> 1=Enables MCA reporting of DDR3 DRAM correctable ECC errors which are detected in the NB.

### MSR0000\_0411 MC4 Machine Check Status (MC4\_STATUS)

Not-same-for-all. Cold reset: 0000\_0000\_0000\_0000h.

MSR0000\_0411[63:32] is an alias of D18F3x4C, and MSR0000\_0411[31:0] is an alias of D18F3x48. Only one of these registers exists in multi-core devices; see 3.1.1 [Northbridge MSRs In Multi-Core Products]. Accessibility of this register by non-NBC cores is affected by D18F3x44[NbMcaToMstCpuEn]. See MSRC001\_0015[McStatusWrEn] for information on writing to this register. See 2.14.1 [Machine Check Architecture] for machine check architecture background.

Table 205 describes each error type. Table 206 and Table 207 describe the error codes and status register settings for each error type.

Bits	Description
63	<b>Val: valid.</b> See: MSR0000_0401[Val].
62	<b>Overflow: error overflow.</b> See: MSR0000_0401[Overflow].
61	<b>UC: error uncorrected.</b> See: MSR0000_0401[UC].
60	<b>En: error enable.</b> See: MSR0000_0401[En].
59	<b>MiscV: miscellaneous error register valid.</b> See: MSR0000_0401[MiscV]. 1=Valid thresholding in MSR0000_0413 or MSRC000_0408.
58	<b>AddrV: error address valid.</b> See: MSR0000_0401[AddrV].
57	<b>PCC: processor context corrupt.</b> See: MSR0000_0401[PCC].

56	<b>ErrCoreIdVal: error core ID is valid.</b> Read-write; set-by-hardware. 1=The ErrCoreId field is valid.
55	Reserved.
54:47	<b>Syndrome[7:0].</b> Read-write. Syndrome[15:0] = {Syndrome[15:8], Syndrome[7:0]}. The syndrome bits when an ECC error is detected. See <a href="#">Table 207</a> Valid Syndrome column for which bits are valid for each error.
46	<b>CECC: correctable ECC error.</b> Read-write; Updated-by-hardware. 1=The error was a correctable ECC error according to the restrictions of the ECC algorithm. UC indicates whether the error was actually corrected by the processor.
45	<b>UECC: uncorrectable ECC error.</b> Read-write; Updated-by-hardware. 1=The error was an uncorrectable ECC error according to the restrictions of the ECC algorithm. UC indicates whether the error was actually corrected by the processor.
44	<b>Deferred: deferred error.</b> Read-write; Updated-by-hardware. 1=A deferred error was created. A deferred error is the result of an uncorrectable data error which did not immediately cause a processor exception; the data is poisoned and an exception is deferred until the data is consumed.
43:42	Reserved.
41	<b>SubLink: sublink.</b> Read-write; set-by-hardware. For errors associated with a link, this bit indicates if the error was associated with the upper or lower byte of the link. 0=Sublink [7:0]. 1=Sublink [15:8].
40	<b>Scrub: error detected on a scrub.</b> Read-write; Set-by-hardware.
39:37	Reserved.
36	<b>Link.</b> Read-write; set-by-hardware. For errors associated with a link, this field indicates that the link was associated with the error.
35:32	<b>ErrCoreId: error associated with core N.</b> Read-write; updated-by-hardware. When ErrCoreIdVal=1 this field indicates which core within the processor is associated with the error; Otherwise this field is reserved. All values greater than <a href="#">D18F5x84</a> [CmpCap] are reserved.
31:24	Reserved.
23:21	Reserved.
20:16	<b>ErrorCodeExt: extended error code.</b> Read-write; Updated-by-hardware. See <a href="#">MSR0000_0401</a> [ErrorCodeExt]. See <a href="#">Table 206</a> for values.
15:0	<b>ErrorCode: error code.</b> Read-write; Updated-by-hardware. See <a href="#">2.14.1.5</a> [Error Code].

**Table 205: MC4 Error Descriptions**

Error Type	Description	CTL <sup>1</sup>	ETG <sup>2</sup>	EAC <sup>4</sup>
Sync Error	Link-defined sync error packets detected on link. The NB floods its outgoing links with sync packets after detecting a sync packet on an incoming link independent of the state of the control bits.	SyncPktEn	L	D
Master Abort	Master abort seen as result of link operation. Reasons for this error include requests to non-existent addresses. 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	L	D
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	L	D

**Table 205: MC4 Error Descriptions**

Error Type	Description	CTL <sup>1</sup>	ETG <sup>2</sup>	EAC <sup>4</sup>
RMW Error	An atomic read-modify-write (RMW) command was received from an IO 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 the control bit.	AtomicRMWEn	L	D
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 node or device which failed to respond.	WDTRptEn	L	D
DRAM ECC Error	A DRAM ECC error detected.	CECCEn, UECCEn	D	D
DRAM CRC Error	A DRAM CRC error was detected.	CCRCEn, UCRCEn	D	D
Link Data Error	Data error detected on link. If enabled for reporting and the request is sourced from a core, then PCC is set. (If not enabled for reporting, PCC is not set. If configured to allow an error response to be returned to the core, this could allow error containment to a scope smaller than the entire system.)	McaUsPwDatErrEn, CpPktDatEn	L	D
Protocol Error	Protocol error detected by link. These errors are distinguished from each other by the value in <a href="#">MSR0000_0412[ErrAddr]</a> . See <a href="#">Table 209</a> .  For protocol errors, the system cannot continue operation. Protocol errors can be caused by other subcomponents than the one reporting the error. For diagnosis, collect and examine MCA registers from other banks, cores, and processors in the system.	NbIntProtEn	L <sup>3</sup>	D
NB Array Error	A parity error was detected in the NB internal arrays.	NbArrayParEn	-	D
L2 complex Data Error	NB received a data error from a core and this error could not be contained. For the cause of the data error, examine the core MCA registers for deferred errors. This error may occur for the following types of data writes: <ul style="list-style-type: none"> <li>• APIC</li> <li>• Configuration space (IO and MMIO)</li> </ul> For these errors, sync flood will occur if <a href="#">D18F3x180[SyncFloodOnCpuLeakErr]</a> is set.	McaCpuDatErrEn	-	D

1. CTL: See [MSR0000\\_0410](#).

2. ETG: error threshold group. See [2.14.1.7 \[Error Thresholding\]](#).

- L=Link.
- D=DRAM.

3. The error thresholding group is Link if link protocol error; none for non-link protocol error.

4. EAC: D=Error action taken if detected. E=Error action taken if MCA bank enabled.

The NB is capable of reporting the following errors:

**Table 206: MC4 Error Signatures, Part 1**

Error Type	ErrorCode-Ext	Error Code					
		Type	PP	T	RRRR	II/TT	LL
Reserved	00h	-	-	-	-	-	-
Reserved	01h	-	-	-	-	-	-
Sync Error	02h	BUS	OBS	0	GEN	GEN	LG
Mst Abort	03h	BUS	SRC/OBS	0	RD/WR	MEM/IO <sup>1</sup>	LG
Tgt Abort	04h	BUS	SRC/OBS	0	RD/WR	MEM/IO <sup>1</sup>	LG
RMW Error	06h	BUS	OBS	0	GEN	IO	LG
WDT Error	07h	BUS	GEN	1	GEN	GEN	LG
ECC Error	08h	BUS	SRC/RES	0	RD/WR	MEM	LG
Link Data Error	0Ah	BUS	SRC/OBS	0	RD/WR/DWR	MEM/IO	LG
NB Protocol Error	0Bh		OBS	0	GEN	GEN	LG
NB Array Error	0Ch		OBS	0	GEN	GEN	LG
L2 complex Data Error	19h	MEM	-	-	WR	Data	LG

1. Indicates the type of link attached to the reporting NB, not the instruction type. MEM indicates coherent link, IO indicates IO link.

**Table 207: MC4 Error Signatures, Part 2**

Error Type	UC	AddrV	PCC	Syndrome Valid	CECC	UECC	Deferred	Scrub	Link	Err CoreId
Sync Error	1	0	1	-	0	0	0	0	Y	-
Mst Abort	1	1	Core <sup>10</sup>	-	0	0	0	0	Y	Y
Tgt Abort	1	1	Core <sup>10</sup>	-	0	0	0	0	Y	Y
RMW Error	1	1	0	-	0	0	0	0	Y	-
WDT Error	1	0 <sup>1</sup>	1	-	0	0	0	0	-	-
ECC Error	MS <sup>6</sup>	1	MS <sup>6</sup> & Core	15:0	~MS <sup>6</sup>	MS <sup>6</sup>	0	1/0	-	-
Link Data Error	~Deferred <sup>11</sup>	1	0	-	0	0	0/1	0	Y	-
NB Protocol Error	1	1/0 <sup>2</sup>	1	-	0	0	0	0	Y	-
NB Array Error	~Deferred <sup>11</sup>	1 <sup>4</sup>	~Deferred <sup>11</sup>	-	0	0	0/1	0	-	-
L2 complex Data Error	1	0	1	-	0	0	0	0	-	Y

**Table 207: MC4 Error Signatures, Part 2**

Error Type	UC	AddrV	PCC	Syndrome Valid	CECC	UECC	Deferred	Scrub	Link	Err CoreId
1. See Table 213 [Format of MSR0000_0412[ErrAddr[47:1]] for Watchdog Timer Errors]. 2. See Table 209 [Format of MSR0000_0412[ErrAddr[47:1]] for Protocol Errors]. 3. See Table 212 [Valid Values for ArrayErrorType]. 4. MS: multi-symbol. 1=Multi-symbol. 0=Not multi-symbol. 5. Core: source is core. 1=Source is core. 0=Source is not core. 6. Deferred: error is deferred. 1=Error is deferred. 0=Error is not deferred.										

**MSR0000\_0412 MC4 Machine Check Address (MC4\_ADDR)**

IF (D18F3x44[NbMcaToMstCpuEn] && !NBC) THEN RAZ; Per-node; not-same-for-all. ELSE Read-write; Per-node; not-same-for-all. ENDIF. Cold reset: 0000\_0000\_0000\_0000h. See 2.14.1 [Machine Check Architecture]. MSR0000\_0412[31:0] is an alias of D18F3x50. MSR0000\_0412[63:32] is an alias of D18F3x54. Only one of these registers exists in multi-core devices; see 3.1.1 [Northbridge MSRs In Multi-Core Products]. ErrAddr[47:1] carries supplemental information associated with a machine check error, generally the address being accessed. Accessibility of this register by non-NBC cores is affected by D18F3x44[NbMcaToMstCpuEn]. The format of ErrAddr[47:1] is a function of MSR0000\_0411[ErrorCodeExt]; See ErrAddr[47:1].

Bits	Description
63:48	Reserved. Value: 00h.
47:32	<b>ErrAddr[47:32]: Error Address Bits[47:32]</b> . See: ErrAddr[31:1].
31:1	<b>ErrAddr[31:1]: Error Address Bits[31:1]</b> . ErrAddr[47:0] = {MSR0000_0412[ErrAddr[47:32]], MSR0000_0412[ErrAddr[31:1]], MSR0000_0412[ErrAddr[0]]}. See the tables below for the encoding.
0	<b>ErrAddr[0]: Error Address Bit[0]</b> .

The register format depends on the type of error being logged:

- Protocol errors contain the error reason code, may contain the physical address, and are formatted according to Table 209.
- NB array errors indicate the array in error, and are formatted according to Table 211.
- NB Watchdog timer errors depend on the mode selected by D18F3x180[McaLogErrAddrWdtErr], and the format is indicated by D18F3x4C[AddrV]. If D18F3x4C[AddrV] is indicated, errors are formatted according to Table 208. If D18F3x4C[AddrV] is not indicated, errors are formatted according to Table 213.
- All other NB errors which indicate D18F3x4C[AddrV] are formatted according to Table 208.

Table 208: Format of MSR0000\_0412[ErrAddr[47:1]] for All Other Errors

Bits	Description
47:1	<b>PhysAddr[47:1]</b> .

Table 209: Format of MSR0000\_0412[ErrAddr[47:1]] for Protocol Errors

Bits	Description
47:6	<b>PhysAddr[47:6]</b> . Valid of (MSR0000_0411[AddrV]==1); otherwise reserved.
5:1	<b>ProtocolErrorType</b> . See Table 210 [Valid Values for ProtocolErrorType].



**Table 210: Valid Values** for ProtocolErrorType

Bits	Description
00h	Link: SRQ Read Response without matching request
01h	Link: Probe Response without matching request
02h	Link: TgtDone without matching request
03h	Link: TgtStart without matching request
04h	Link: Command buffer overflow
05h	Link: Data buffer overflow
06h	Link: Link retry packet count acknowledge overflow
07h	Link: Data command in the middle of a data transfer
08h	Link: Link address extension command followed by a packet other than a command with address.
09h	Link: A specific coherent-only packet from a CPU was issued to an IO link.
0Ah	Link: A command with invalid encoding was received. This error occurs when: <ol style="list-style-type: none"> <li>Any invalid command is received (including a command with no valid encoding or a coherent link command over an IO link or vice versa) while not in retry mode.</li> <li>Any illegal command is received in which the CRC is correct while in retry mode (including any upstream broadcast command (link command encoding = 11101xb)).</li> </ol>
0Bh	Link: Link CTL deassertion occurred when a data phase was not pending. This error condition may only occur when error-retry mode is not enabled (if it is enabled, this condition triggers a retry).
0Fh-0Ch	Reserved
1Fh-10h	Reserved

Table 211: Format of [MSR0000\\_0412](#)[ErrAddr[47:1]] for NB Array Errors

Bits	Description
47:6	Reserved.
5:1	<b>ArrayErrorType</b> . See <a href="#">Table 212 [Valid Values for ArrayErrorType]</a> .

Table 212: **Valid Values** for ArrayErrorType

Bits	Description
00h	SRA: System request address.
01h	SRD: System request data.
02h	SPB: System packet buffer.
03h	MCD: Memory controller data.
04h	MPB: Memory packet buffer.
05h	LPB0: Link 0 packet buffer.
08h-06h	Reserved.
09h	MPBC: Memory controller command packet buffer.
0Ah	MCDBM: Memory controller byte mask.
0Bh	MCACAM: Memory controller address array.

Table 212: **Valid Values** for ArrayErrorType

Bits	Description
0Ch	DMAP: Extended DRAM address map.
0Dh	MMAP: Extended MMIO address map.
0Eh	X86MAP: Extended PCI/IO address map.
0Fh	CFGMAP: Extended config address map.
17h-10h	Reserved.
18h	SRIMCTRTE: SRI/MCT extended routing table.
1Ch-19h	Reserved
1Dh	TCB: TCB array.
1Fh-1Eh	Reserved

Table 213: Format of MSR0000\_0412[ErrAddr[47:1]] for Watchdog Timer Errors

Bits	Description																
47:40	<b>CoreId.</b> Indicates the core ID if the SourcePointer specifies Core. <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>07h-00h</td> <td>CoreId</td> </tr> <tr> <td>FFh-08h</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Description	07h-00h	CoreId	FFh-08h	Reserved										
Bits	Description																
07h-00h	CoreId																
FFh-08h	Reserved																
39:36	<b>SystemResponseCount.</b> This field records unspecified, implementation-specific information.																
35:31	<b>WaitCode.</b> records unspecified, implementation-specific information (all zeroes means no waiting condition).																
30	<b>WaitForPostedWrite.</b>																
29:27	<b>DestinationNode.</b> Records the Node ID of the node addressed by the transaction.																
26:25	<b>DestinationUnit.</b> <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Core</td> </tr> <tr> <td>01b</td> <td>Extended Core</td> </tr> <tr> <td>10b</td> <td>Memory Controller</td> </tr> <tr> <td>11b</td> <td>Host</td> </tr> </tbody> </table>	Bits	Description	00b	Core	01b	Extended Core	10b	Memory Controller	11b	Host						
Bits	Description																
00b	Core																
01b	Extended Core																
10b	Memory Controller																
11b	Host																
24:22	<b>SourceNode.</b> Records the Node ID of the node originating the transaction.																
21:20	<b>SourceUnit.</b> (same encoding as Destination Unit)																
19:15	<b>SourcePointer.</b> Identifies crossbar source: <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00h</td> <td>SRI HostBridge</td> </tr> <tr> <td>03h-01h</td> <td>Reserved</td> </tr> <tr> <td>04h</td> <td>Core. See CoreId.</td> </tr> <tr> <td>07h-05h</td> <td>Reserved</td> </tr> <tr> <td>08h</td> <td>Memory controller.</td> </tr> <tr> <td>0Fh-09h</td> <td>Reserved</td> </tr> <tr> <td>1Fh-10h</td> <td>Link. Link HH; sublink N (where N=0b for ganged links). All unused codes are reserved.</td> </tr> </tbody> </table>	Bits	Description	00h	SRI HostBridge	03h-01h	Reserved	04h	Core. See CoreId.	07h-05h	Reserved	08h	Memory controller.	0Fh-09h	Reserved	1Fh-10h	Link. Link HH; sublink N (where N=0b for ganged links). All unused codes are reserved.
Bits	Description																
00h	SRI HostBridge																
03h-01h	Reserved																
04h	Core. See CoreId.																
07h-05h	Reserved																
08h	Memory controller.																
0Fh-09h	Reserved																
1Fh-10h	Link. Link HH; sublink N (where N=0b for ganged links). All unused codes are reserved.																

**Table 213:** Format of `MSR0000_0412`[ErrAddr[47:1]] for Watchdog Timer Errors

Bits	Description
14:11	<b>SrqEntryState.</b> Records unspecified, implementation-specific information (all zeroes means idle).
10:7	<b>OpType.</b> Records unspecified, implementation-specific information.
6:1	<b>LinkCommand.</b> When the NB WDT expires, the link command of the transaction that timed out is captured here. This field is encoded identically to the “Code” field for link transactions defined in the link specification.

**MSR0000\_0413 NB Machine Check Misc 4 (DRAM Thresholding) 0 (MC4\_MISC0)**

`MSR0000_0413` is the first of the NB machine check miscellaneous registers. `MSR0000_0413` is associated with the DRAM error type. To see the remaining NB machine check miscellaneous registers, refer to `MSR0000_0408`. Only one of these registers exists in multi-core devices; see 3.1.1 [Northbridge MSRs In Multi-Core Products].

Bits	Description						
63	<b>Valid.</b> IF (D18F3x44[NbMcaToMstCpuEn] && !NBC) THEN RAZ; <b>Per-node</b> ; not-same-for-all. ELSIF (MSRC001_0015[McStatusWrEn]) THEN Read-write; <b>Per-node</b> ; not-same-for-all. ELSE Read-only; <b>Per-node</b> ; not-same-for-all. Reset: 1. ENDIF. 1=The CntP field is present.						
62	<b>CntP: counter present.</b> IF (D18F3x44[NbMcaToMstCpuEn] && !NBC) THEN RAZ; <b>Per-node</b> ; not-same-for-all. ELSIF (MSRC001_0015[McStatusWrEn]) THEN Read-write; <b>Per-node</b> ; not-same-for-all. ELSE Read-only; <b>Per-node</b> ; not-same-for-all. Reset: 1. ENDIF. 1=A valid threshold counter is present.						
61	<b>Locked.</b> IF (D18F3x44[NbMcaToMstCpuEn] && !NBC) THEN RAZ; <b>Per-node</b> ; not-same-for-all. ELSIF (MSRC001_0015[McStatusWrEn]) THEN Read-write; <b>Per-node</b> ; not-same-for-all. ELSE Read-only; <b>Per-node</b> ; not-same-for-all. Reset: 0. ENDIF. BIOS: IF (IntType==10b) THEN 1. ELSE 0. ENDIF. 1=Writes to bits [55:32] of this register are ignored. Set by BIOS to indicate that this register is not available for OS use. When MSRC001_0015[McStatusWrEn] is set, MSR writes to this register update all bits, regardless of the state of the Locked bit.						
60:56	Reserved.						
55:52	<b>LvtOffset: LVT offset.</b> IF (D18F3x44[NbMcaToMstCpuEn] && !NBC) THEN RAZ; <b>Per-node</b> ; not-same-for-all. ELSIF (MSRC001_0015[McStatusWrEn]   ~MSR0000_0413[Locked]) THEN Read-write; <b>Per-node</b> ; not-same-for-all. ELSE Read-only; <b>Per-node</b> ; not-same-for-all. Reset: 0h. ENDIF. Specifies the address of the LVT entry in the APIC registers as follows: LVT address = (LvtOffset shifted left 4 bits) + 500h (see APIC[530:500]). <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>3h-0h</td> <td>See APIC[530:500].</td> </tr> <tr> <td>Fh-4h</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Description	3h-0h	See APIC[530:500].	Fh-4h	Reserved
Bits	Description						
3h-0h	See APIC[530:500].						
Fh-4h	Reserved						
51	<b>CntEn: counter enable.</b> IF (D18F3x44[NbMcaToMstCpuEn] && !NBC) THEN RAZ; <b>Per-node</b> ; not-same-for-all. ELSIF (MSRC001_0015[McStatusWrEn]   ~Locked) THEN Read-write; <b>Per-node</b> ; not-same-for-all. ELSE Read-only; <b>Per-node</b> ; not-same-for-all. Reset: 0. ENDIF. 1=Count thresholding errors. See 2.14.1.7 [Error Thresholding].						

50:49	<p><b>IntType: interrupt type.</b> IF (D18F3x44[NbMcaToMstCpuEn] &amp;&amp; !NBC) THEN RAZ; <b>Per-node</b>; not-same-for-all. ELSIF (MSRC001_0015[McStatusWrEn]   ~MSR0000_0413[Locked]) THEN Read-write; <b>Per-node</b>; not-same-for-all. ELSE Read-only; <b>Per-node</b>; not-same-for-all. Cold reset: 0. ENDIF. Specifies the type of interrupt signaled when Ovrflw is set.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>No Interrupt.</td> </tr> <tr> <td>01b</td> <td>APIC. APIC based interrupt (see LvtOffset above) to all cores.</td> </tr> <tr> <td>10b</td> <td>SMI. SMI trigger event (always routed to CpuCoreNum 0, as defined in 2.4.4 [Processor Cores and Downcoring]); see 2.4.10.2.3 [SMI Sources And Delivery].</td> </tr> <tr> <td>11b</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Description	00b	No Interrupt.	01b	APIC. APIC based interrupt (see LvtOffset above) to all cores.	10b	SMI. SMI trigger event (always routed to CpuCoreNum 0, as defined in 2.4.4 [Processor Cores and Downcoring]); see 2.4.10.2.3 [SMI Sources And Delivery].	11b	Reserved
Bits	Description										
00b	No Interrupt.										
01b	APIC. APIC based interrupt (see LvtOffset above) to all cores.										
10b	SMI. SMI trigger event (always routed to CpuCoreNum 0, as defined in 2.4.4 [Processor Cores and Downcoring]); see 2.4.10.2.3 [SMI Sources And Delivery].										
11b	Reserved										
48	<p><b>Ovrflw: overflow.</b> IF (D18F3x44[NbMcaToMstCpuEn] &amp;&amp; !NBC) THEN RAZ; <b>Per-node</b>; not-same-for-all. ELSIF (MSRC001_0015[McStatusWrEn]   ~Locked) THEN Read-write; <b>Per-node</b>; not-same-for-all; set-by-hardware. ELSE Read-only; <b>Per-node</b>; not-same-for-all; set-by-hardware. Cold reset: 0. ENDIF. Set by hardware when ErrCnt transitions from FFEh to FFFh. When this bit is set, the interrupt selected by the IntType field is generated.</p>										
47:44	Reserved.										
43:32	<p><b>ErrCnt: error counter.</b> IF (D18F3x44[NbMcaToMstCpuEn] &amp;&amp; !NBC) THEN RAZ; <b>Per-node</b>; not-same-for-all. ELSIF (MSRC001_0015[McStatusWrEn]   ~Locked) THEN Read-write; <b>Per-node</b>; not-same-for-all; updated-by-hardware. ELSE Read-only; <b>Per-node</b>; not-same-for-all; updated-by-hardware. Cold reset: 0. ENDIF. Written by software to set the starting value of the error counter. Incremented by hardware when errors are logged. Saturates at FFFh (no rollover). The threshold value, written by software, is (FFFh - the desired error count (the number of errors necessary in order for an interrupt to be taken)).</p>										
31:24	<p><b>BlkPtr: Block pointer for additional MISC registers.</b> Read-only. Value: 01h. 01h=Extended MC4_MISC MSR block is valid. See <a href="#">MSRC000_0408</a>.</p>										
23:0	Reserved.										

### MSR0000\_0414 MC5 Machine Check Control (MC5\_CTL)

Read-write. Reset: 0000\_0000\_0000\_0000h. See 2.14.1 [Machine Check Architecture]. See [MSRC001\\_0049 \[FR Machine Check Control Mask \(MC5\\_CTL\\_MASK\)\]](#).

Bits	Description
63:1	Unused.
0	<b>WDT: core watchdog timer.</b> See <a href="#">MSRC001_0074 [CPU Watchdog Timer (CpuWdtCfg)]</a> .

### MSR0000\_0415 MC5 Machine Check Status (MC5\_STATUS)

Cold reset: 0000\_0000\_0000\_0000h. See 2.14.1 [Machine Check Architecture]. See [MSRC001\\_0015\[McStatusWrEn\]](#). [Table 214](#) describes each error type. [Table 215](#) describes the error codes and status register settings for each error type.

Bits	Description
63	<b>Val: error valid.</b> See: <a href="#">MSR0000_0401[Val]</a> .
62	<b>Overflow: error overflow.</b> See: <a href="#">MSR0000_0401[Overflow]</a> .

61	<b>UC: error uncorrected.</b> See: <a href="#">MSR0000_0401[UC]</a> .
60	<b>En: error enable.</b> See: <a href="#">MSR0000_0401[En]</a> .
59	<b>MiscV: miscellaneous error register valid.</b> Read-write; Updated-by-hardware. See: <a href="#">MSR0000_0401[MiscV]</a> . 1= <a href="#">MSR0000_0417</a> is valid.
58	<b>AddrV: error address valid.</b> See: <a href="#">MSR0000_0401[AddrV]</a> . 1=Valid address in <a href="#">MSR0000_0416</a> .
57	<b>PCC: processor context corrupt.</b> See: <a href="#">MSR0000_0401[PCC]</a> .
56:16	Reserved.
15:0	<b>ErrorCode: error code.</b> Read-write; Updated-by-hardware. See <a href="#">2.14.1.5 [Error Code]</a> See <a href="#">Table 215</a> .

**Table 214: MC5 Error Descriptions**

Error Type	Error Sub-type	Description	CTL <sup>1</sup>	EAC <sup>3</sup>
WDT error	-	The WDT timer has expired. See <a href="#">MSRC001_0074 [CPU Watchdog Timer (CpuWdtCfg)]</a> .	WDT	E

1. CTL: See [MSR0000\\_0414](#).  
 2. CID: core ID. All EX errors are reported to the affected core; see [2.14.1.3 \[Error Detection, Action, Logging, and Reporting\]](#).  
 3. EAC: D=Error action taken if detected. E=Error action taken if MCA bank enabled. See [2.14.1.3 \[Error Detection, Action, Logging, and Reporting\]](#).

**Table 215: MC5 Error Signatures**

Error Type	Error Sub-Type	ErrorCodeExt	Error Code						UC	ADDR	MISCV	PCC	CECC	UECC
			Type	UU/PP	TT	RRRR	II/TT	LL						
WDT error	-	00h	BUS	GEN	1	GEN	GEN	LG	1	1	1	1	0	0

1. Causes shutdown.  
 2. Causes shutdown if UC=1.  
 3. STATQ sets PCC=1 if a store retired in last 3 cycles or FpTokenRet bits mismatch.

**MSR0000\_0416 MC5 Machine Check Address (MC5\_ADDR)**

Read-write; Updated-by-hardware. Cold reset: 0000\_0000\_0000\_0000h. The MCi\_ADDR register contains valid data if indicated by MCi\_STATUS[AddrV]. See [2.14.1 \[Machine Check Architecture\]](#). The register format depends on the type of error being logged.

Bits	Description
63:0	<b>ADDR.</b> See <a href="#">Table 216</a> .

The following tables define the address register as a function of error type.

**Table 216: MC5 Address Register**

Error Type	Error Sub-Type	Bits	Description
WDT	- (ErrorCodeExt=00h)	63:48	Reserved
		48:0	<b>LogAddr[48:0]</b> . Logical canonical address of the next instruction after the last instruction retired.

**MSR0000\_0417 MC5 Machine Check Miscellaneous (MC5\_MISC)**

Cold reset: 0000\_0000\_0000\_0000h.

Bits	Description
63:8	Reserved.
7:0	<b>FrCompl.</b> Read-write.

**3.19 MSRs - MSRC000\_0xxx****MSRC000\_0080 Extended Feature Enable (EFER)**

SKINIT Execution: 0000\_0000\_0000\_0000h.

Bits	Description
63:16	MBZ.
15	MBZ.
14	<b>FFXSE: fast FXSAVE/FRSTOR enable.</b> Read-write. Reset: 0. 1=Enables the fast FXSAVE/FRSTOR mechanism. A 64-bit operating system may enable the fast FXSAVE/FRSTOR mechanism if ( <code>CPUID Fn8000_0001_EDX[FFXSR]==1</code> ). This bit is set once by the operating system and its value is not changed afterwards.
13	<b>LMSLE: long mode segment limit enable.</b> Read-write. Reset: 0. 1=Enables the long mode segment limit check mechanism.
12	<b>SVME: secure virtual machine (SVM) enable.</b> IF ( <code>MSRC001_0114[SvmeDisable]==1</code> ) THEN MBZ. ELSE Read-write. ENDIF. Reset: 0. 1=SVM features are enabled.
11	<b>NXE: no-execute page enable.</b> Read-write. Reset: 0. 1=The no-execute page protection feature is enabled.
10	<b>LMA: long mode active.</b> Read-only. Reset: 0. 1=Indicates that long mode is active.
9	MBZ.
8	<b>LME: long mode enable.</b> Read-write. Reset: 0. 1=Long mode is enabled.
7:1	RAZ.
0	<b>SYSCALL: system call extension enable.</b> Read-write. Reset: 0. 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 (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	<b>SysRetSel: SYSRET CS and SS.</b> Read-write.
47:32	<b>SysCallSel: SYSCALL CS and SS.</b> Read-write.
31:0	<b>Target: SYSCALL target address.</b> Read-write.

**MSRC000\_0082 Long Mode SYSCALL Target Address (STAR64)**

Reset: 0000\_0000\_0000\_0000h.

Bits	Description
63:0	<b>LSTAR: long mode target address.</b> 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 (STARCOMPAT)**

Reset: 0000\_0000\_0000\_0000h.

Bits	Description
63:0	<b>CSTAR: compatibility mode target address.</b> 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 (SYSCALL\_FLAG\_MASK)**

Bits	Description
63:32	RAZ.
31:0	<b>Mask: SYSCALL flag mask.</b> Read-write. Reset: 0000_0000h. 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 (FS\_BASE)**

Reset: 0000\_0000\_0000\_0000h.

Bits	Description
63:0	<b>FSBase: expanded FS segment base.</b> Read-write; not-same-for-all. 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 (GS\_BASE)**

Reset: 0000\_0000\_0000\_0000h.

Bits	Description
63:0	<b>GSBase: expanded GS segment base.</b> Read-write; not-same-for-all. 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 (KernelGSbase)**

Reset: 0000\_0000\_0000\_0000h.

Bits	Description
63:0	<b>KernelGSBase: kernel data structure pointer.</b> 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 (TSC\_AUX)**

Reset: 0000\_0000\_0000\_0000h.



Bits	Description
63:32	Reserved.
31:0	<b>TscAux: auxiliary time stamp counter data.</b> 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.

### **MSRC000\_0104 Time Stamp Counter Ratio (TscRateMsr)**

**MSRC000\_0104 [Time Stamp Counter Ratio (TscRateMsr)]** allows the hypervisor to control the guest's view of the Time Stamp Counter. It provides a multiplier that scales the value returned when **MSR0000\_0010[TSC]** and **MSR0000\_00E7[MPERF]** are read by a guest running under virtualization. This allows the hypervisor to provide a consistent TSC and MPERF rate for a guest process when moving that process between cores that have a differing P0 rate. The TSC Ratio MSR does not affect the value read from the TSC and MPERF MSRs when read when in host mode or when virtualization is not being used or when accessed by code executed in system management mode (SMM) unless the SMM code is executed within a guest container. The TSC Ratio value does not affect the rate of the underlying TSC and MPERF counters, or the value that gets written to the TSC and MPERF MSRs counters on a write by either the host or the guest. The TSC Ratio MSR contains a fixed-point number in 8.32 format, which is 8 bits of integer and 32 bits of fraction. This number is the ratio of the desired P0 frequency to the P0 frequency of the core. The reset value of the TSC Ratio MSR is 1.0, which results in a guest frequency matches the core P0 frequency.

Bits	Description
63:40	MBZ.
39:32	<b>TscRateMsrInt: time stamp counter rate integer.</b> Read-write. Reset: 01h. Specifies the integer part of the MSR TSC ratio value.
31:0	<b>TscRateMsrFrac: time stamp counter rate fraction.</b> Read-write. Reset: 0000_0000h. Specifies the fractional part of the MSR TSC ratio value.

### **MSRC000\_0408 NB Machine Check Misc 4 (Link Thresholding) 1 (MC4\_MISC1)**

Per-node. **MSRC000\_0408** is associated with the link error type. See 2.14.1.7 [Error Thresholding]. Accessibility of this register by non-NBC cores is affected by **D18F3x44[NbMcaToMstCpuEn]**.

Bits	Description
63	<b>Valid.</b> IF (D18F3x44[NbMcaToMstCpuEn] && !NBC) THEN RAZ; <b>Per-node</b> ; not-same-for-all. ELSIF (MSRC001_0015[McStatusWrEn]) THEN Read-write; <b>Per-node</b> ; not-same-for-all. ELSE Read-only; <b>Per-node</b> ; not-same-for-all. Reset: 1. ENDIF. 1=The CntP field is present.
62	<b>CntP: counter present.</b> IF (D18F3x44[NbMcaToMstCpuEn] && !NBC) THEN RAZ; <b>Per-node</b> ; not-same-for-all. ELSIF (MSRC001_0015[McStatusWrEn]) THEN Read-write; <b>Per-node</b> ; not-same-for-all. ELSE Read-only; <b>Per-node</b> ; not-same-for-all. Reset: 1. ENDIF. 1=A valid threshold counter is present.
61	<b>Locked.</b> IF (D18F3x44[NbMcaToMstCpuEn] && !NBC) THEN RAZ; <b>Per-node</b> ; not-same-for-all. ELSIF (MSRC001_0015[McStatusWrEn]) THEN Read-write; <b>Per-node</b> ; not-same-for-all. ELSE Read-only; <b>Per-node</b> ; not-same-for-all. Reset: 0. ENDIF. BIOS: IF (IntType==10b) THEN 1. ELSE 0. ENDIF. 1=Writes to bits [55:32] of this register are ignored. Set by BIOS to indicate that this register is not available for OS use. When MSRC001_0015[McStatusWrEn] is set, MSR writes to this register update all bits, regardless of the state of the Locked bit.

60:56	Reserved.										
55:52	<p><b>LvtOffset: LVT offset.</b> IF (D18F3x44[NbMcaToMstCpuEn] &amp;&amp; !NBC) THEN RAZ; <b>Per-node</b>; not-same-for-all. ELSIF (MSRC001_0015[McStatusWrEn]   ~MSRC000_0408[Locked]) THEN Read-write; <b>Per-node</b>; not-same-for-all. ELSE Read-only; <b>Per-node</b>; not-same-for-all. Reset: 0h. ENDIF. Specifies the address of the LVT entry in the APIC registers as follows: LVT address = (LvtOffset shifted left 4 bits) + 500h (see APIC[530:500]).</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>3h-0h</td> <td>See APIC[530:500].</td> </tr> <tr> <td>Fh-4h</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Description	3h-0h	See APIC[530:500].	Fh-4h	Reserved				
Bits	Description										
3h-0h	See APIC[530:500].										
Fh-4h	Reserved										
51	<p><b>CntEn: counter enable.</b> IF (D18F3x44[NbMcaToMstCpuEn] &amp;&amp; !NBC) THEN RAZ; <b>Per-node</b>; not-same-for-all. ELSIF (MSRC001_0015[McStatusWrEn]   ~Locked) THEN Read-write; <b>Per-node</b>; not-same-for-all. ELSE Read-only; <b>Per-node</b>; not-same-for-all. Reset: 0. ENDIF. 1=Count thresholding errors. See 2.14.1.7 [Error Thresholding].</p>										
50:49	<p><b>IntType: interrupt type.</b> IF (D18F3x44[NbMcaToMstCpuEn] &amp;&amp; !NBC) THEN RAZ; <b>Per-node</b>; not-same-for-all. ELSIF (MSRC001_0015[McStatusWrEn]   ~MSRC000_0408[Locked]) THEN Read-write; <b>Per-node</b>; not-same-for-all. ELSE Read-only; <b>Per-node</b>; not-same-for-all. Cold reset: 0. ENDIF. Specifies the type of interrupt signaled when Ovrflw is set.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>No Interrupt.</td> </tr> <tr> <td>01b</td> <td>APIC. APIC based interrupt (see LvtOffset above) to all cores.</td> </tr> <tr> <td>10b</td> <td>SMI. SMI trigger event (always routed to CpuCoreNum 0, as defined in 2.4.4 [Processor Cores and Downcoring]); see 2.4.8.2.3 [SMI Sources And Delivery].</td> </tr> <tr> <td>11b</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Description	00b	No Interrupt.	01b	APIC. APIC based interrupt (see LvtOffset above) to all cores.	10b	SMI. SMI trigger event (always routed to CpuCoreNum 0, as defined in 2.4.4 [Processor Cores and Downcoring]); see 2.4.8.2.3 [SMI Sources And Delivery].	11b	Reserved
Bits	Description										
00b	No Interrupt.										
01b	APIC. APIC based interrupt (see LvtOffset above) to all cores.										
10b	SMI. SMI trigger event (always routed to CpuCoreNum 0, as defined in 2.4.4 [Processor Cores and Downcoring]); see 2.4.8.2.3 [SMI Sources And Delivery].										
11b	Reserved										
48	<p><b>Ovrflw: overflow.</b> IF (D18F3x44[NbMcaToMstCpuEn] &amp;&amp; !NBC) THEN RAZ; <b>Per-node</b>; not-same-for-all. ELSIF (MSRC001_0015[McStatusWrEn]   ~Locked) THEN Read-write; <b>Per-node</b>; not-same-for-all; set-by-hardware. ELSE Read-only; <b>Per-node</b>; not-same-for-all; set-by-hardware. Cold reset: 0. ENDIF. Set by hardware when ErrCnt transitions from FFEh to FFFh. When this bit is set, the interrupt selected by the IntType field is generated.</p>										
47:44	Reserved.										
43:32	<p><b>ErrCnt: error counter.</b> IF (D18F3x44[NbMcaToMstCpuEn] &amp;&amp; !NBC) THEN RAZ; <b>Per-node</b>; not-same-for-all. ELSIF (MSRC001_0015[McStatusWrEn]   ~Locked) THEN Read-write; <b>Per-node</b>; not-same-for-all; updated-by-hardware. ELSE Read-only; <b>Per-node</b>; not-same-for-all; updated-by-hardware. Cold reset: 0. ENDIF. Written by software to set the starting value of the error counter. Incremented by hardware when errors are logged. Saturates at FFFh (no rollover). The threshold value, written by software, is (FFFh - the desired error count (the number of errors necessary in order for an interrupt to be taken)).</p>										
31:0	Reserved.										

**MSRC000\_0409 Reserved**

Bits	Description
63:0	RAZ.

**MSRC000\_040[F:A] Reserved**

---

Bits	Description
63:0	RAZ.

**3.20 MSRs - MSRC001\_0xxx****MSRC001\_00[03:00] Performance Event Select (PERF\_CTL[3:0])**

Reset: 0000\_0000\_0000\_0000h. See 2.6.1 [Performance Monitor Counters].

Table 217: Register Mapping for MSRC001\_00[03:00]

Register	Function
MSRC001_0000	Counter 0
MSRC001_0001	Counter 1
MSRC001_0002	Counter 2
MSRC001_0003	Counter 3

Bits	Description										
63:42	Reserved.										
41	<b>HostOnly: host only counter.</b> Read-write. 1=Events are only counted when the processor is in host mode.										
40	<b>GuestOnly: guest only counter.</b> Read-write. 1=Events are only counted when the processor is in guest mode.										
39:36	Reserved.										
35:32	<b>EventSelect[11:8]: performance event select.</b> Read-write. See: EventSelect[7:0].										
31:24	<b>CntMask: counter mask.</b> Read-write. Controls the number of events counted per clock cycle. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00h</td> <td>The corresponding PERF_CTR[3:0] register is incremented by the number of events occurring in a clock cycle. Maximum number of events in one cycle is 3.</td> </tr> <tr> <td>03h-01h</td> <td>When Inv = 0, the corresponding PERF_CTR[3:0] register is incremented by 1 if the number of events occurring in a clock cycle is greater than or equal to the CntMask value.</td> </tr> <tr> <td></td> <td>When Inv = 1, the corresponding PERF_CTR[3:0] register is incremented by 1 if the number of events occurring in a clock cycle is less than CntMask value.</td> </tr> <tr> <td>FFh-04h</td> <td>Reserved.</td> </tr> </tbody> </table>	Bits	Definition	00h	The corresponding PERF_CTR[3:0] register is incremented by the number of events occurring in a clock cycle. Maximum number of events in one cycle is 3.	03h-01h	When Inv = 0, the corresponding PERF_CTR[3:0] register is incremented by 1 if the number of events occurring in a clock cycle is greater than or equal to the CntMask value.		When Inv = 1, the corresponding PERF_CTR[3:0] register is incremented by 1 if the number of events occurring in a clock cycle is less than CntMask value.	FFh-04h	Reserved.
Bits	Definition										
00h	The corresponding PERF_CTR[3:0] register is incremented by the number of events occurring in a clock cycle. Maximum number of events in one cycle is 3.										
03h-01h	When Inv = 0, the corresponding PERF_CTR[3:0] register is incremented by 1 if the number of events occurring in a clock cycle is greater than or equal to the CntMask value.										
	When Inv = 1, the corresponding PERF_CTR[3:0] register is incremented by 1 if the number of events occurring in a clock cycle is less than CntMask value.										
FFh-04h	Reserved.										
23	<b>Inv: invert counter mask.</b> Read-write. See CntMask.										
22	<b>En: enable performance counter.</b> Read-write. 1= Performance event counter is enabled.										
21	Reserved.										
20	<b>Int: enable APIC interrupt.</b> Read-write. 1=APIC performance counter LVT interrupt is enabled to generate an interrupt when the performance counter overflows.										
19	Reserved.										
18	<b>Edge: edge detect.</b> Read-write. 0=Level detect. 1=Edge detect. 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.										
17	<b>OS: OS mode.</b> Read-write. 1=Events are only counted when CPL=0.										
16	<b>User: user mode.</b> Read-write. 1=Events only counted when CPL>0.										

15:8	<b>UnitMask: event qualification.</b> 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	<b>EventSelect[7:0]: event select.</b> 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_CNT[3:0] register. The events are specified in section 3.22 [Core Performance Counter Events]. Some events are reserved; when a reserved event is selected, the results are undefined.

**MSRC001\_00[07:04] Performance Event Counter (PERF\_CTR[3:0])**

The core provides four 48-bit performance counters. Each counter can monitor a different event specified by MSRC001\_00[03:00] [Performance Event Select (PERF\_CTL[3:0])]. 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\_CNT[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.

Table 218: Register Mapping for MSRC001\_00[07:04]

Register	Function
MSRC001_0004	Counter 0
MSRC001_0005	Counter 1
MSRC001_0006	Counter 2
MSRC001_0007	Counter 3

Bits	Description
63:48	RAZ.
47:0	<b>CTR: performance counter value.</b> Read-write. Reset: 0. Returns the current value of the event counter.

### MSRC001\_0010 System Configuration (SYS\_CFG)

Bits	Description
63:23	Reserved.
22	<b>Tom2ForceMemTypeWB: top of memory 2 memory type write back.</b> Read-write. Reset: 0. 1=The default memory type of memory between 4GB and TOM2 is write back instead of the memory type defined by <a href="#">MSR0000_02FF[MemType]</a> . For this bit to have any effect, <a href="#">MSR0000_02FF[MtrrDefTypeEn]</a> must be 1. MTRRs and PAT can be used to override this memory type.
21	<b>MtrrTom2En: MTRR top of memory 2 enable.</b> Read-write. Reset: 0. 0= <a href="#">MSRC001_001D [Top Of Memory 2 (TOM2)]</a> is disabled. 1=This register is enabled. See <a href="#">D0F0x64_x19[TomEn]</a> .
20	<b>MtrrVarDramEn: MTRR variable DRAM enable.</b> Read-write. Reset: 0. BIOS: 1. 0= <a href="#">MSRC001_001A [Top Of Memory (TOP_MEM)]</a> and IORRs are disabled. 1=These registers are enabled.
19	<b>MtrrFixDramModEn: MTRR fixed RdDram and WrDram modification enable.</b> Read-write. Reset: 0. Controls access to <a href="#">MSR0000_02[6F:68,59:58,50][RdDram, WrDram]</a> . 0=Access type is MBZ; writing 00b does not change the hidden value of <a href="#">MSR0000_02[6F:68,59:58,50][RdDram, WrDram]</a> . 1=Access type is Read-write. BIOS: This bit should be set to 1 during BIOS initialization of the fixed MTRRs, then cleared to 0 for operation.
18	<b>MtrrFixDramEn: MTRR fixed RdDram and WrDram attributes enable.</b> Read-write. Reset: 0. BIOS: 1. 1=Enables the RdDram and WrDram attributes in <a href="#">MSR0000_02[6F:68,59:58,50]</a> .
17	<b>SysUcLockEn: system lock command enable.</b> Read-write. Reset: 1. 0=Coherent fabric does not support bus lock transactions. This bit must be set on multi-core processors.
16	Reserved.
15:0	Reserved.

### MSRC001\_0015 Hardware Configuration (HWCR)

Bits	Description
63:32	Reserved.
31:30	Reserved.
29	Reserved.
28	Reserved.
27	Reserved.

26	<b>EffFreqCntMwait: effective frequency counting during mwait.</b> Read-write. Reset: 0. Specifies whether <a href="#">MSR0000_00E7 [Max Performance Frequency Clock Count (MPERF)]</a> and <a href="#">MSR0000_00E8 [Actual Performance Frequency Clock Count (APERF)]</a> increment while the core is in the monitor event pending state. 0=The registers do not increment. 1=The registers increment. See <a href="#">2.5.3.3 [Effective Frequency]</a> .
25	<b>CpbDis: core performance boost disable.</b> Read-write. Reset: 0. Specifies whether core performance boost is requested to be enabled or disabled. 0=CPB is requested to be enabled. 1=CPB is disabled. See <a href="#">2.5.9 [Application Power Management (APM)]</a> . If core performance boost is disabled while a core is in a boosted P-state, the core will automatically transition to the highest performance non-boosted P-state.
24	<b>TscFreqSel: TSC frequency select.</b> Read-only. Reset: 1. 1=The TSC increments at the P0 frequency. This field uses software P-state numbering. See <a href="#">2.5.3.1.1.1 [Software P-state Numbering]</a> .
23	<b>ForceRdWrSzPrb: force probes for RdSized and WrSized.</b> Read-write. Reset: 0. A read returns a 1 if this field is set on any core of the node. 1=Forces probes on read-sized and write-sized transactions, except those that are display refresh.
22	Reserved.
21	<b>MisAlignSseDis: misaligned SSE mode disable.</b> Read-write. Reset: 0. 1=Disables misaligned SSE mode. If this is set, then <a href="#">CPUID Fn8000_0001_ECX[MisAlignSse]</a> is 0.
20	<b>IoCfgGpFault: IO-space configuration causes a GP fault.</b> Read-write. Reset: 0. 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 <a href="#">MSRC001_0054 [IO Trap Control (SMI_ON_IO_TRAP_CTL_STS)]</a> .
19	Reserved.
18	<b>McStatusWrEn: machine check status write enable.</b> Read-write. Reset: 0. McStatusWrEn can be used to debug machine check exception and interrupt handlers. See <a href="#">2.14.3 [Error Injection and Simulation]</a> . See <a href="#">2.14.1 [Machine Check Architecture]</a> . <ul style="list-style-type: none"> <li>• 1=Writes by software to <a href="#">MCI_STATUS</a> (see <a href="#">2.14.1 [Machine Check Architecture]</a>) do not cause general protection faults; such writes update all implemented bits in these registers.</li> <li>• 0=<a href="#">MCI_STATUS</a> registers are readable; writing a non-zero pattern to these registers causes a general protection fault.</li> <li>• The <a href="#">MCI_STATUS</a> registers are: <a href="#">MSR0000_0401</a>, <a href="#">MSR0000_0405</a>, <a href="#">MSR0000_0409</a>, <a href="#">MSR0000_040D</a>, <a href="#">MSR0000_0411</a>, <a href="#">MSR0000_0415</a>. McStatusWrEn does not affect the writability of <a href="#">MSR0000_0001</a>; <a href="#">MSR0000_0001</a> is always writable.</li> </ul>
17	<b>Wrap32Dis: 32-bit address wrap disable.</b> Read-write. Reset: 0. 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 <a href="#">MSRC000_0100 [FS Base (FS_BASE)]</a> and <a href="#">MSRC000_0101 [GS Base (GS_BASE)]</a> . Then it would address $\pm 2$ 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.
16:15	Reserved.
14	<b>RsmSpCycDis: RSM special bus cycle disable.</b> IF <a href="#">MSRC001_0015[SmmLock]</a> THEN Read-only ELSE Read-write ENDIF. Reset: 0. 0=A link special bus cycle, SMIACK, is generated on a resume from SMI.



13	<b>SmiSpCycDis: SMI special bus cycle disable.</b> IF <a href="#">MSRC001_0015</a> [SmmLock] THEN Read-only ELSE Read-write ENDIF. Reset: 0. 0=A link special bus cycle, SMIACK, is generated when an SMI interrupt is taken.
12	Reserved.
11	Reserved.
10	<b>MonMwaitUserEn: MONITOR/MWAIT user mode enable.</b> Read-write. Reset: 0. 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	<b>MonMwaitDis: MONITOR and MWAIT disable.</b> Read-write. Reset: 0. 1=The MONITOR and MWAIT opcodes become invalid. This affects what is reported back through <a href="#">CPUID Fn0000_0001_ECX</a> [Monitor].
8	<b>IgneEm: IGNNE port emulation enable.</b> Read-write. Reset: 0. 1=Enable emulation of IGNNE port.
7:6	Reserved.
5	Reserved.
4	<b>INVDWBINVD: INVD to WBINVD conversion.</b> Read-write. Reset: 0. 1=Convert INVD to WBINVD. BIOS: See <a href="#">2.3.3 [Using L2 Cache as General Storage During Boot]</a> . This bit is required to be set for normal operation when two or more cores of a L2 complex are enabled, and thus share the L2 cache.
3	<b>TlbCacheDis: cacheable memory disable.</b> Read-write. Reset: 0. 1=Disable performance improvement that assumes that the PML4, PDP, PDE and PTE entries are in cacheable WB DRAM. Operating systems that maintain page tables in any other memory type must set the TlbCacheDis bit to insure proper operation. <ul style="list-style-type: none"> <li>TlbCacheDis does not override the memory type specified by the SMM ASeg and TSeg memory regions. See <a href="#">2.4.8.2.7 [The Protected ASeg and TSeg Areas]</a>.</li> </ul>
2	Reserved.
1	Reserved.
0	<b>SmmLock: SMM code lock.</b> Read; write-1-only. Reset: 0. 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. See <a href="#">2.4.8.2.9 [Locking SMM]</a> .

### **MSRC001\_00[18,16] IO Range Base (IORR\_BASE[1:0])**

Reset: 0000\_0000\_0000\_0000h. 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 core 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].

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. See [2.4.6.1.2 \[Determining The Access Destination for Core Accesses\]](#).

Bits	Description
63:40	RAZ.
39:12	<b>PhyBase: physical base address.</b> Read-write.
11:5	RAZ.
4	<b>RdMem: read from memory.</b> Read-write. 1=Read accesses to the range are directed to system memory. 0=Read accesses to the range are directed to IO.
3	<b>WrMem: write to memory.</b> Read-write. 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 Mask (IORR\_MASK[1:0])

Reset: 0000\_0000\_0000\_0000h. See [MSRC001\\_00\[18,16\]](#).

Bits	Description
63:40	RAZ.
39:12	<b>PhyMask: physical address mask.</b> Read-write.
11	<b>Valid.</b> Read-write. 1=The pair of registers that specifies an IORR range is valid.
10:0	RAZ.

### MSRC001\_001A Top Of Memory (TOP\_MEM)

Reset: 0000\_0000\_0000\_0000h.

Bits	Description
63:40	RAZ.
39:23	<b>TOM[39:23]: top of memory.</b> Read-write. 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 <a href="#">2.4.6 [System Address Map]</a> and <a href="#">2.9.11 [DRAM CC6/PC6 Storage]</a> .
22:0	RAZ.

### MSRC001\_001D Top Of Memory 2 (TOM2)

Reset: 0000\_0000\_0000\_0000h.

Bits	Description
63:40	RAZ.
39:23	<b>TOM2[39:23]: second top of memory.</b> Read-write. 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 <a href="#">2.4.6 [System Address Map]</a> and <a href="#">2.9.11 [DRAM CC6/PC6 Storage]</a> . This register is enabled by <a href="#">MSRC001_0010[MtrrTom2En]</a> .
22:0	RAZ.

**MSRC001\_001F Northbridge Configuration 1 (NB\_CFG1)**

Read-write; Per-node. Only one of these registers exists in multi-core devices; see 3.1.1 [Northbridge MSRs In Multi-Core Products]. MSRC001\_001F[31:0] is an alias of D18F3x88. MSRC001\_001F[63:32] is an alias of D18F3x8C.

Bits	Description
63	Reserved.
62	<b>DisStpClkAbortFlush.</b> Reset: 0. BIOS: 1. 1=Disable aborting flush for core when the other core has a pending architectural interrupt.
61:56	Reserved.
55	<b>EnaDiv1CpuLowPwr.</b> Reset: 0. BIOS: 1. Enables power management actions in the core even when the requested clock divisor is /1. Normally a /1 clock divisor does not generate power management actions.
54	<b>InitApicIdCpuIdLo.</b> Reset: 0. BIOS: 1. 0=Reserved. 1=Selects the format for ApicId; see APIC20.
53:52	Reserved.
51	<b>DisDatFwdVic.</b> Reset: 0. BIOS: 1. 1=Disables data forwarding from victims to reads.
50	<b>DisOrderRdRsp.</b> Reset: 0. 1=Disables ordered responses to IO link read requests.
49:47	Reserved.
46	<b>EnableCf8ExtCfg: enable CF8 extended configuration cycles.</b> Reset: 0. 1=Allows the IO configuration space access method, IOCF8 and IOCFC, to be used to generate extended configuration cycles by enabling IOCF8[27:24].
45	<b>DisUsSysMgtReqToNcHt: disable upstream system management request to link.</b> Reset: 0. 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).
44:37	Reserved.
36	<b>DisDatMsk: disable data mask.</b> Reset: 0. BIOS: IF (DataMaskMbType!=1) THEN 1 ELSE 0 ENDIF. 1=Disables DRAM data masking function; all write requests that are less than one cacheline, a DRAM read is performed before writing the data. Data masking is supported in ECC mode; the NB performs a minimum write size of 16B.
35:32	Reserved.
31	<b>DisCohLdtCfg: disable coherent link configuration accesses.</b> Reset: 0. 1=Disables automatic routing of PCI configuration accesses to the processor configuration registers; PCI configuration space accesses which fall within the hard-coded range reserved for processor configuration-space registers are instead routed to the IO link specified by D18F1x[1DC:1D0,EC:E0] [Configuration Map]. This can be used to effectively hide the configuration registers from software. It can also be used to provide a means for an external chip to route processor configuration accesses according to a scheme other than the hard-coded version. When used, this bit needs to be set on all processors in a system. PCI configuration accesses should not be generated if this bit is not set on all processors.
30:28	Reserved.

27	<b>DisDramScrub.</b> Reset: 0. 1=Disable DRAM ECC scrubbing; this overrides the settings in <a href="#">D18F3x58 [Scrub Rate Control]</a> , <a href="#">D18F3x5C [DRAM Scrub Address Low]</a> .
26:19	Reserved.
18	<b>DisCstateBoostBlockPstateUp.</b> Read-write. Reset: 0. BIOS: <a href="#">D18F5x88[EnCstateBoostBlockCC6Exit]</a> . 1=Allow cores that are waking up out of a non-C0 C-state to transition to the last requested Pstate without having to wait for cores in the boosted P-state to transition out of the boosted P-state.
17:0	Reserved.

### MSRC001\_0022 Machine Check Exception Redirection

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
63:32	Reserved.
31:10	Reserved.
9	<b>RedirSmiEn.</b> Read-write. 1=Redirect MCEs (that are directed to this core) to generate an SMI-trigger IO cycle via <a href="#">MSRC001_0056</a> . The status is stored in <a href="#">SMMFEC4[MceRedirSts]</a> .
8	<b>RedirVecEn.</b> 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	<b>RedirVector.</b> Read-write. See RedirVecEn.

### MSRC001\_00[35:30] Processor Name String

Reset: 0000\_0000\_0000\_0000h. BIOS: [Table 220](#). These registers holds the CPUID name string in ASCII. The state of these registers are returned by CPUID instructions, [CPUID Fn8000\\_000\[4:2\]\\_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.

Table 219: [Register Mapping](#) for [MSRC001\\_00\[35:30\]](#)

Register	Function
MSRC001_0030	Characters 7-0
MSRC001_0031	Characters 15-8
MSRC001_0032	Characters 23-16
MSRC001_0033	Characters 31-24
MSRC001_0034	Characters 39-32
MSRC001_0035	Characters 47-40

See [D18F5x194](#) for the access method to [D18F5x198\\_x\[B:0\]](#).

**Table 220: BIOS Recommendations for MSRC001\_00[35:30]**

Register	BIOS
MSRC001_0030	{D18F5x198_x1, D18F5x198_x0}
MSRC001_0031	{D18F5x198_x3, D18F5x198_x2}
MSRC001_0032	{D18F5x198_x5, D18F5x198_x4}
MSRC001_0033	{D18F5x198_x7, D18F5x198_x6}
MSRC001_0034	{D18F5x198_x9, D18F5x198_x8}
MSRC001_0035	{D18F5x198_xB, D18F5x198_xA}

Bits	Description
63:0	<b>CpuNameString</b> . Read-write.

**MSRC001\_003E Hardware Thermal Control (HTC)**

Only one of these registers exists in multi-core devices; see 3.1.1 [Northbridge MSRs In Multi-Core Products]. Uses hardware P-state numbering. See 2.5.3.1.1.2 [Hardware P-state Numbering].

Bits	Description
63:32	Reserved.
31	Reserved.
30:28	<b>HtcPstateLimit: HTC P-state limit select.</b> MSRC001_003E[HtcPstateLimit] is an alias of D18F3x64[HtcPstateLimit]. MSRC001_003E[HtcPstateLimit] is an alias of D18F3x64[HtcPstateLimit].
27:24	<b>HtcHystLmt: HTC hysteresis.</b> MSRC001_003E[HtcHystLmt] is an alias of D18F3x64[HtcHystLmt].
23	<b>HtcSlewSel: HTC slew-controlled temperature select.</b> MSRC001_003E[HtcSlewSel] is an alias of D18F3x64[HtcSlewSel].
22:16	<b>HtcTmpLmt: HTC temperature limit.</b> MSRC001_003E[HtcTmpLmt] is an alias of D18F3x64[HtcTmpLmt].
15:8	Reserved.
7	<b>PslApicLoEn: P-state limit lower value change APIC interrupt enable.</b> MSRC001_003E[PslApicLoEn] is an alias of D18F3x64[PslApicLoEn].
6	<b>PslApicHiEn: P-state limit higher value change APIC interrupt enable.</b> MSRC001_003E[PslApicHiEn] is an alias of D18F3x64[PslApicHiEn].
5	<b>HtcActSts: HTC-active status.</b> MSRC001_003E[HtcActSts] is an alias of D18F3x64[HtcActSts].
4	<b>HtcAct: HTC-active state.</b> MSRC001_003E[HtcAct] is an alias of D18F3x64[HtcAct].
3:1	Reserved.
0	<b>HtcEn: HTC enable.</b> MSRC001_003E[HtcEn] is an alias of D18F3x64[HtcEn].

**MSRC001\_0044 DC Machine Check Control Mask (MC0\_CTL\_MASK)**

Read-write. Reset: 0000\_0000\_0000\_0000h. BIOS: 0000\_0000\_0000\_0000h. See 2.14.1 [Machine Check Architecture]. See MSR0000\_0400 [MC0 Machine Check Control (MC0\_CTL)].

Bits	Description
63:12	Reserved.
11	<b>SRDE_ALLMsk: all system read data errors mask.</b> Read-write. See DDPMsk.
10	<b>SRDETMsk: read data errors on TLB reload mask.</b> Read-write. See DDPMsk.
9	<b>SRDESMsk: read data errors on store mask.</b> Read-write. See DDPMsk.
8	<b>SRDELMsk: read data errors on load mask.</b> Read-write. See DDPMsk.
7	<b>TLB2MMsk: L2DTLB multi-hit mask.</b> Read-write. 1=Disable detection. See DDPMsk.
6	<b>TLB2PMsk: L2DTLB parity mask.</b> Read-write. 1=Disable detection. See DDPMsk.
5	<b>TLB1MMsk: L1DTLB multi-hit mask.</b> Read-write. 1=Disable detection. See DDPMsk.
4	Reserved.
3	<b>DTPMsk: tag array parity errors mask.</b> Read-write. See DDPMsk.
2	<b>DDPMsk: data array parity errors mask.</b> Read-write. 1=Disable error logging in <a href="#">MSR0000_0401 [MC0 Machine Check Status (MC0_STATUS)]</a> and <a href="#">MSR0000_0402 [MC0 Machine Check Address (MC0_ADDR)]</a> for errors represented by this bit. This bit also disables error detection, and prevents error responses. See <a href="#">2.14.1 [Machine Check Architecture]</a> and <a href="#">MSR0000_0400 [MC0 Machine Check Control (MC0_CTL)]</a> .
1:0	Reserved.

#### **MSRC001\_0045 IC Machine Check Control Mask (MC1\_CTL\_MASK)**

Reset: 0000\_0000\_0000\_0000h. BIOS: 0000\_0000\_0000\_0000h. See [2.14.1 \[Machine Check Architecture\]](#). See [MSR0000\\_0404 \[MC1 Machine Check Control \(MC1\\_CTL\)\]](#).

Bits	Description
63:10	Reserved.
9	<b>SRDEMsk: read data errors mask.</b> Read-write. See: IDPMsk.
8:7	Reserved.
6	<b>TLBPMsk: TLB parity errors mask.</b> Read-write. See: IDPMsk.
5	Reserved.
4	<b>ISTPMsk: snoop tag array parity errors mask.</b> Read-write. See: IDPMsk.
3	<b>ITPMsk: tag array parity errors mask.</b> Read-write. See: IDPMsk.
2	<b>IDPMsk: data array parity errors mask.</b> Read-write. 1=Disable error logging in <a href="#">MSR0000_0405 [MC1 Machine Check Status (MC1_STATUS)]</a> and <a href="#">MSR0000_0406 [MC1 Machine Check Address (MC1_ADDR)]</a> for errors represented by this bit. This bit also disables error detection, and prevents error responses. See <a href="#">2.14.1 [Machine Check Architecture]</a> and <a href="#">MSR0000_0404 [MC1 Machine Check Control (MC1_CTL)]</a> .
1	<b>IVPMsk: victim parity error mask.</b> Read-write. 1=Disable logging victim parity errors.
0	<b>PRPEMsk: ME patch RAM parity error mask.</b> Read-write. 1=Disable logging patch ram parity errors.

#### **MSRC001\_0046 BU Machine Check Control Mask (MC2\_CTL\_MASK)**

Read-write. Reset: 0000\_0000\_0000\_0000h. BIOS: 0000\_0000\_0000\_0000h. See [2.14.1 \[Machine Check](#)

Architecture]. See MSR0000\_0408 [MC2 Machine Check Control (MC2\_CTL)].

Bits	Description
63:14	Reserved.
13	<b>AttrParity: report attribute parity errors.</b>
12:9	Reserved.
8	<b>DataUncor: report uncorrectable data errors.</b>
7	<b>DataCor: report correctable data errors.</b>
6	<b>DataParity: report data parity errors.</b>
5	<b>TagUncor: report uncorrectable tag errors.</b>
4	<b>TagCor: report correctable tag errors.</b>
3	Reserved.
2	<b>NbFill: NB read response error due to a fill.</b>
1	<b>NbRead: NB read response error due to a write/victim.</b>
0	Reserved.

#### MSRC001\_0047 Reserved (MC3\_CTL\_MASK)

Reset: 0000\_0000\_0000\_0000h.

Bits	Description
63:0	Reserved.

#### MSRC001\_0048 NB Machine Check Control Mask (MC4\_CTL\_MASK)

IF (D18F3x44[NbMcaToMstCpuEn] && !NBC) THEN RAZ. ELSE Read-write; Per-node. Reset: 0000\_0000\_0400\_0000h. BIOS: 0000\_0000\_0400\_0000h. ENDIF. The format of MC4\_CTL\_MASK corresponds to MSR0000\_0410 [MC4 Machine Check Control (MC4\_CTL)]. For each defined bit position, 1=Disable logging. Only one of these registers exists in multi-core devices; see 3.1.1 [Northbridge MSRs In Multi-Core Products]. See 2.14.1 [Machine Check Architecture]. Accessibility of this register by non-NBC cores is affected by D18F3x44[NbMcaToMstCpuEn]. See MSR0000\_0410 [MC4 Machine Check Control (MC4\_CTL)].

Bits	Description
63:32	Reserved.
31	<b>McaCpuDatErrEn.</b>
30	Reserved.
29	<b>UCRCEn.</b>
28	<b>CCRCEn.</b>
27	Reserved.
26	<b>NbArrayParEn.</b>
25	<b>UsPwDatErrEn.</b>
24:18	Reserved.

17	<b>CpPktDatEn.</b>
16	<b>NbIntProtEn.</b>
15:13	Reserved.
12	<b>WDTRptEn.</b>
11	<b>AtomicRMWEn.</b>
10	Reserved.
9	<b>TgtAbortEn.</b>
8	<b>MstrAbortEn.</b>
7:6	Reserved.
5	<b>SyncPktEn.</b>
4:2	Reserved.
1	<b>UECCEn.</b>
0	<b>CECCEn.</b>

#### **MSRC001\_0049 FR Machine Check Control Mask (MC5\_CTL\_MASK)**

Reset: 0000\_0000\_0000\_0000h. BIOS: 0000\_0000\_0000\_0000h. See 2.14.1 [Machine Check Architecture]. See MSR0000\_0414 [MC5 Machine Check Control (MC5\_CTL)].

Bits	Description
63:1	Reserved.
0	<b>CPUWDTMsk: CPU watchdog timer mask.</b> Read-write. 1=Disable error logging in MSR0000_0415 [MC5 Machine Check Status (MC5_STATUS)] and MSR0000_0416 [MC5 Machine Check Address (MC5_ADDR)] for errors represented by this bit. This bit also disables error detection, and prevents error responses. See 2.14.1 [Machine Check Architecture] and MSR0000_0414 [MC5 Machine Check Control (MC5_CTL)].

#### **MSRC001\_00[53:50] IO Trap (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 performed 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.

The conditional GP fault described by [MSRC001\\_0015](#)[IoCfgGpFault] takes priority over this trap.

Table 221: [Register Mapping](#) for [MSRC001\\_00](#)[53:50]

Register	Function
MSRC001_0050	Range 0
MSRC001_0051	Range 1
MSRC001_0052	Range 2
MSRC001_0053	Range 3

Bits	Description
63	<b>SmiOnRdEn: enable SMI on IO read.</b> Read-write. 1=Enables SMI generation on a read access.
62	<b>SmiOnWrEn: enable SMI on IO write.</b> Read-write. 1=Enables SMI generation on a write access.
61	<b>ConfigSmi: configuration space SMI.</b> Read-write. 1=Configuration access. 0=IO access (that is not an IO-space configuration access).
60:56	Reserved.
55:32	<b>SmiMask[23:0].</b> Read-write. SMI IO trap mask. 0=Mask address bit. 1=Do not mask address bit.
31:0	<b>SmiAddr[31:0].</b> Read-write. SMI IO trap address.

#### **MSRC001\_0054 IO Trap Control (SMI\_ON\_IO\_TRAP\_CTL\_STS)**

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	<b>IoTrapEn: IO trap enable.</b> Read-write. Reset: 0. 1=Enable IO and configuration space trapping specified by <a href="#">MSRC001_00</a> [53:50] and <a href="#">MSRC001_0054</a> .
14:8	Reserved.
7	<b>SmiEn3: SMI enable for the trap specified by MSRC001_0053.</b> Read-write. Reset: 0.
6	Reserved.
5	<b>SmiEn2: SMI enable for the trap specified by MSRC001_0052.</b> Read-write. Reset: 0.
4	Reserved.
3	<b>SmiEn1: SMI enable for the trap specified by MSRC001_0051.</b> Read-write. Reset: 0.
2	Reserved.
1	<b>SmiEn0: SMI enable for the trap specified by MSRC001_0050.</b> Read-write. Reset: 0.
0	Reserved.

**MSRC001\_0055 Interrupt Pending**

Reset: 0000\_0000\_0000\_0000h.

Bits	Description
63:0	RAZ.

**MSRC001\_0056 SMI Trigger IO Cycle**

**Not-same-for-all.** Reset: 0000\_0000\_0000\_0000h. See 2.4.8.2.3 [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
63:32	Reserved.
31:27	Reserved.
26	<b>IoRd: IO Read.</b> Read-write. 1=IO read; 0=IO write.
25	<b>IoCycleEn: IO cycle enable.</b> Read-write. 1=The SMI trigger IO cycle is enabled to be generated.
24	Reserved.
23:16	<b>IoData.</b> Read-write.
15:0	<b>IoPortAddress.</b> Read-write.

**MSRC001\_0058 MMIO Configuration Base Address**

**Same-for-all.** See 2.7 [Configuration Space] for a description of MMIO configuration space.

Bits	Description																								
63:40	RAZ.																								
39:20	<p><b>MmioCfgBaseAddr[39:20]: MMIO configuration base address bits[39:20].</b> Read-write. Reset: 0. Specifies the base address of the MMIO configuration range. The size of the MMIO configuration-space address range is specified by BusRange as follows. All lower order undefined bits must be 0.</p> <table border="1"> <thead> <tr> <th>BusRange</th> <th>MmioCfgBaseAddr</th> <th>BusRange</th> <th>MmioCfgBaseAddr</th> </tr> </thead> <tbody> <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> </tbody> </table>	BusRange	MmioCfgBaseAddr	BusRange	MmioCfgBaseAddr	0h	[39:20]	5h	[39:25]	1h	[39:21]	6h	[39:26]	2h	[39:22]	7h	[39:27]	3h	[39:23]	8h	[39:28]	4h	[39:24]	Fh-9h	Reserved
BusRange	MmioCfgBaseAddr	BusRange	MmioCfgBaseAddr																						
0h	[39:20]	5h	[39:25]																						
1h	[39:21]	6h	[39:26]																						
2h	[39:22]	7h	[39:27]																						
3h	[39:23]	8h	[39:28]																						
4h	[39:24]	Fh-9h	Reserved																						
19:6	RAZ.																								

5:2	<b>BusRange: bus range identifier.</b> Read-write. Reset: 0. Specifies the number of buses in the MMIO configuration space range. The size of the MMIO configuration space is 1 MB times the number of buses.																								
	<table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Description</u></th> <th><u>Bits</u></th> <th><u>Description</u></th> </tr> </thead> <tbody> <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> </tbody> </table>	<u>Bits</u>	<u>Description</u>	<u>Bits</u>	<u>Description</u>	0h	1	5h	32	1h	2	6h	64	2h	4	7h	128	3h	8	8h	256	4h	16	Fh-9h	Reserved
<u>Bits</u>	<u>Description</u>	<u>Bits</u>	<u>Description</u>																						
0h	1	5h	32																						
1h	2	6h	64																						
2h	4	7h	128																						
3h	8	8h	256																						
4h	16	Fh-9h	Reserved																						
1	Reserved.																								
0	<b>Enable.</b> Read-write. Reset: 0. 1=MMIO configuration space is enabled.																								

### MSRC001\_0060 BIST Results

Read; GP-write. 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. If (MSRC001\_0060[29:0]=0000\_0000h) then no BIST failures were detected.

Bits	Description
63:32	Reserved.
31:0	<b>BistResults.</b>

### MSRC001\_0061 P-state Current Limit

Read; GP-write; updated-by-hardware. See [2.5.3 \[CPU Power Management\]](#).

Bits	Description
63:7	RAZ.
6:4	<b>PstateMaxVal: P-state maximum value.</b> 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. This field uses software P-state numbering. See <a href="#">2.5.3.1.1.1 [Software P-state Numbering]</a> .
3	RAZ.
2:0	<b>CurPstateLimit: current P-state limit.</b> Specifies the highest-performance non-boosted P-state (lowest value) allowed. CurPstateLimit is always bounded by MSRC001_0061[PstateMaxVal]. Attempts to change the CurPstateLimit to a value greater (lower performance) than MSRC001_0061[PstateMaxVal] leaves CurPstateLimit unchanged. This field uses software P-state numbering. See <a href="#">MSRC001_0071[CurPstateLimit]</a> and <a href="#">2.5.3.1.1.1 [Software P-state Numbering]</a> . CurPstateLimit = D18F3x64[HtcPstateLimit].

**MSRC001\_0062 P-state Control**

Bits	Description
63:3	MBZ.
2:0	<b>PstateCmd: P-state change command.</b> Read-write; <b>Not-same-for-all</b> . Cold reset value varies by product; after a warm reset, value initializes to the P-state the core was in prior to the reset. Writes to this field cause the core to change to the indicated non-boosted P-state number, specified by <a href="#">MSRC001_00[6B:64]</a> . 0=P0, 1=P1, etc. P-state limits are applied to any P-state requests made through this register. Reads from this field return the last written value, regardless of whether any limits are applied. This field uses software P-state numbering. See <a href="#">2.5.3 [CPU Power Management]</a> and <a href="#">2.5.3.1.1.1 [Software P-state Numbering]</a> .

**MSRC001\_0063 P-state Status**

Read; GP-write; Updated-by-hardware.

Bits	Description
63:3	RAZ.
2:0	<b>CurPstate: current P-state.</b> Cold reset: Varies by product. This field provides the frequency component of the current non-boosted P-state of the core (regardless of the source of the P-state change, including <a href="#">MSRC001_0062[PstateCmd]</a> ; see <a href="#">2.5.3.1.5 [Core P-state Transition Behavior]</a> for information on how these interact). 0=P0, 1=P1, etc. The value of this field is updated when the COF transitions to a new value associated with a P-state. This field uses software P-state numbering. See <a href="#">2.5.3 [CPU Power Management]</a> and <a href="#">2.5.3.1.1.1 [Software P-state Numbering]</a> .

**MSRC001\_00[6B:64] P-state [7:0]**

**Per-node.** Cold reset: Varies by product. Each of these registers specify the frequency and voltage associated with each of the core P-states.

Table 222: [Register Mapping](#) for [MSRC001\\_00\[6B:64\]](#)

Register	Function
MSRC001_0064	P-state 0
MSRC001_0065	P-state 1
MSRC001_0066	P-state 2
MSRC001_0067	P-state 3
MSRC001_0068	P-state 4
MSRC001_0069	P-state 5
MSRC001_006A	P-state 6
MSRC001_006B	P-state 7

The CpuVid field in these registers is required to be programmed to the same value in all cores of a processor, but are allowed to be different between processors in a multi-processor system. All other fields in these registers are required to be programmed to the same value in each core of the coherent fabric. See [2.5.3 \[CPU Power Management\]](#).

When [D18F4x15C\[BoostLock\]=1](#), [MSRC001\\_00\[6B:64\]\[CpuVid, CpuDid, CpuFid\]](#) have special write

requirements associated with them.

**Table 223: P-state Definitions**

Term	Definition
CoreCOF	Core current operating frequency in MHz. CoreCOF = 100 * (MSRC001_00[6B:64][CpuFid] + 10h) / (2^MSRC001_00[6B:64][CpuDid]).

Bits	Description										
63	<b>PstateEn.</b> 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.										
62:42	RAZ.										
41:40	<b>IddDiv: current divisor.</b> Read-write. See IddValue.										
39:32	<b>IddValue: current value.</b> Read-write. After a reset, IddDiv and IddValue combine to specify the expected maximum current dissipation of a single core that is in the P-state corresponding to the MSR number. These values are intended to be used to create ACPI-defined _PSS objects (see 2.5.3.1.8.3 [ACPI Processor P-state Objects]) and to perform the 2.5.3.1.7 [Processor-Systemboard Power Delivery Compatibility Check]. 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 are encoded as follows: <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>IddDiv</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>IddValue / 1 A, Range: 0 to 255 A.</td> </tr> <tr> <td>01b</td> <td>IddValue / 10 A, Range: 0 to 25.5 A.</td> </tr> <tr> <td>10b</td> <td>IddValue / 100 A, Range: 0 to 2.55 A.</td> </tr> <tr> <td>11b</td> <td>Reserved</td> </tr> </tbody> </table>	IddDiv	Description	00b	IddValue / 1 A, Range: 0 to 255 A.	01b	IddValue / 10 A, Range: 0 to 25.5 A.	10b	IddValue / 100 A, Range: 0 to 2.55 A.	11b	Reserved
IddDiv	Description										
00b	IddValue / 1 A, Range: 0 to 255 A.										
01b	IddValue / 10 A, Range: 0 to 25.5 A.										
10b	IddValue / 100 A, Range: 0 to 2.55 A.										
11b	Reserved										
31:23	RAZ.										
22	<b>NbPstate: Northbridge P-state.</b> IF (MSRC001_0071[NbPstateDis]) THEN Read-only. ELSE Read-write. ENDIF. 1=Low performance NB P-state. 0=High performance NB P-state. If this bit is set in any given P-state register, then it must also be set in all enabled lower performance P-state registers as well. Equivalent P-states in each core must program this bit to the same value. See 2.5.4.1 [NB P-states] and D18F5x170[NbPstateThreshold, NbPstateLo, NbPstateHi].										
21	RAZ.										
20:17	RAZ.										
16	<b>CpuVid[7]: core VID bit[7].</b> Read-write. Except as required by 2.5.3.1.7 [Processor-Systemboard Power Delivery Compatibility Check], software should not modify this field. See CpuVid[6:0].										
15:9	<b>CpuVid[6:0]: core VID.</b> Read-write. Except as required by 2.5.3.1.7 [Processor-Systemboard Power Delivery Compatibility Check], software should not modify this field. See 2.5.1 [Processor Power Planes And Voltage Control].										

8:6	<p><b>CpuDid: core divisor ID.</b> Read-write. Except as required by <a href="#">2.5.3.1.7 [Processor-Systemboard Power Delivery Compatibility Check]</a>, software should not modify this field. Specifies the core frequency divisor; see CpuFid.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Divide-by 1</td> </tr> <tr> <td>1h</td> <td>Divide-by 2</td> </tr> <tr> <td>2h</td> <td>Divide-by 4</td> </tr> <tr> <td>3h</td> <td>Divide-by 8</td> </tr> <tr> <td>4h</td> <td>Divide-by 16</td> </tr> <tr> <td>7h-5h</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Description	0h	Divide-by 1	1h	Divide-by 2	2h	Divide-by 4	3h	Divide-by 8	4h	Divide-by 16	7h-5h	Reserved
Bits	Description														
0h	Divide-by 1														
1h	Divide-by 2														
2h	Divide-by 4														
3h	Divide-by 8														
4h	Divide-by 16														
7h-5h	Reserved														
5:0	<p><b>CpuFid[5:0]: core frequency ID.</b> Read-write. Except as required by <a href="#">2.5.3.1.7 [Processor-Systemboard Power Delivery Compatibility Check]</a>, software should not modify this field. Specifies the core frequency multiplier. The core COF is a function of CpuFid and CpuDid, and defined by <a href="#">CoreCOF</a>.</p>														

**MSRC001\_0070 COFVID Control**

Cold reset: Product-specific. There is one register implemented for each core. This register includes several fields that are identical to [MSRC001\\_00\[6B:64\]](#). It is controlled by hardware for P-state transitions. It may also be used by software to directly control the current COF or VID.

Accesses to this register that result in invalid COFs or VIDs are ignored. See [2.5.3 \[CPU Power Management\]](#).

Bits	Description
63:32	RAZ.
31:24	<b>NbVid: Northbridge VID.</b> IF ( <a href="#">MSRC001_0071[NbPstateDis]</a> ) THEN Read-only. ELSE Read-write. ENDIF. See <a href="#">D18F5x16[C:0][NbVid]</a> .
23	RAZ.
22	<b>NbPstate: Northbridge P-state.</b> IF ( <a href="#">MSRC001_0071[NbPstateDis]</a> ) THEN Read-only. ELSE Read-write. ENDIF. See <a href="#">MSRC001_00[6B:64][NbPstate]</a> .
21	RAZ.
20	<b>CpuVid[7].</b> Read-write. See CpuVid[6:0].
19	RAZ.
18:16	<b>PstateId: P-state identifier.</b> Read-write. This field is required to provide the P-state number that is associated with the values of the other fields in this register. This value is used by the logic to determine if the P-state is increasing or decreasing. This field uses hardware P-state numbering. See <a href="#">2.5.3.1.1.2 [Hardware P-state Numbering]</a> .
15:9	<b>CpuVid[6:0]: core VID.</b> Read-write. See <a href="#">MSRC001_00[6B:64][CpuVid]</a> . CpuVid[7:0] = {CpuVid[7], CpuVid[6:0]}.
8:6	<b>CpuDid: core divisor ID.</b> Read-write. See <a href="#">MSRC001_00[6B:64][CpuDid]</a> . The PstateId field must be updated to cause a new CpuDid value to take effect.
5:0	<b>CpuFid[5:0]: core frequency ID.</b> Read-write. See <a href="#">MSRC001_00[6B:64][CpuFid]</a> . The PstateId field must be updated to cause a new CpuFid value to take effect.

**MSRC001\_0071 COFVID Status**

See 2.5.3 [CPU Power Management].

Bits	Description
63:59	<b>MaxNbCof: maximum NB COF.</b> Read-only. Cold reset: Product-specific. Specifies the maximum NB COF supported by the processor. If MaxNbCof is greater than zero, the maximum frequency is 100 MHz * MaxNbCof; if MaxNbCof = 00h, then there is no frequency limit. Any attempt to change the NB COF to a frequency greater than specified by this field is ignored.
58:56	<b>CurPstateLimit: current P-state limit.</b> Read-only; updated-by-hardware. Provides the current lowest-performance P-state limit number. This register uses hardware P-state numbering. See <a href="#">MSRC001_0061[CurPstateLimit]</a> and 2.5.3.1.1.2 [Hardware P-state Numbering].
55	Reserved.
54:49	<b>MaxCpuCof: maximum core COF.</b> Read-only. Cold reset: Product-specific. Specifies the maximum CPU COF supported by the processor. The maximum frequency is 100 MHz * MaxCpuCof, if MaxCpuCof is greater than zero; if MaxCpuCof = 00h, then there is no frequency limit. Any attempt to change a CPU COF to a frequency greater than specified by this field is ignored.
48:35	Reserved.
34:32	<b>StartupPstate: startup P-state number.</b> Read-only. Cold reset: Product-specific. Specifies the cold reset VID, FID and DID for the core based on the P-state number selected. StartupPstate uses hardware P-state numbering. See <a href="#">MSRC001_00[6B:64]</a> and 2.5.3.1.1.2 [Hardware P-state Numbering].
31:24	<b>CurNbVid[7:0]: current NB VID.</b> Read-only; updated-by-hardware. Cold reset: Product-specific. This field specifies the current VDDNB voltage. <a href="#">MSRC001_0071[CurNbVid[7:0]]</a> is an alias of <a href="#">D18F5x174[CurNbVid[7:0]]</a> .
23	<b>NbPstateDis: NB P-states disabled.</b> Value: <a href="#">D18F5x174[NbPstateDis]</a> . <a href="#">MSRC001_0071[NbPstateDis]</a> is an alias of <a href="#">D18F5x174[NbPstateDis]</a> . 0=NB P-state frequency and voltage changes are supported. See <a href="#">D18F5x170[SwNbPstateLoDis, NbPstateDisOnP0]</a> . 1=NB P-state frequency and voltage changes are disabled.
22	Reserved.
21	Reserved.
20	<b>CurCpuVid[7].</b> Read-only; updated-by-hardware; not-same-for-all. Cold reset: Product-specific. See <a href="#">CurCpuVid[6:0]</a> .
19	Reserved.
18:16	<b>CurPstate: current P-state.</b> Read-only; updated-by-hardware; not-same-for-all. Cold reset: Product-specific. Specifies the current P-state requested by the core. This field uses hardware P-state numbering. See <a href="#">MSRC001_0063[CurPstate]</a> and 2.5.3.1.1.2 [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	<b>CurCpuVid[6:0]: current core VID.</b> Read-only; updated-by-hardware; not-same-for-all. Cold reset: Product-specific. $CurCpuVid = \{CurCpuVid[7], CurCpuVid[6:0]\}$ . This field specifies the current VDD voltage.

8:6	<b>CurCpuDid: current core divisor ID.</b> Read-only; updated-by-hardware. Cold reset: Product-specific. Specifies the current CpuDid of the core. See <a href="#">MSRC001_00[6B:64]</a> . When a P-state change is requested, the value in this field is updated once all required frequency and voltage transitions are completed.
5:0	<b>CurCpuFid: current core frequency ID.</b> Read-only; updated-by-hardware; not-same-for-all. Cold reset: Product-specific. Specifies the current CpuFid of the core. See <a href="#">MSRC001_00[6B:64]</a> . 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 Base Address

Reset: 0000\_0000\_0000\_0000h.

Bits	Description
63:32	Reserved.
31:16	Reserved.
15:0	<b>CstateAddr: C-state address.</b> Read-write. 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. Writing values greater than FFF8h into this field result in undefined behavior. All other values cause the core to trap IO addresses CstateAddr through CstateAddr+7. See <a href="#">2.5.3.2.2 [C-state Request Interface]</a> , <a href="#">D18F4x11[C:8]</a> , and <a href="#">D18F4x11C</a> .

### MSRC001\_0074 CPU Watchdog Timer (CpuWdtCfg)

Read-write; **Same-for-all**. 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 ([D18F3x44 \[MCA NB Configuration\]](#)) 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	<p><b>CpuWdtCountSel: CPU watchdog timer count select.</b> CpuWdtCountSel and CpuWdtTimeBase together specify the time period required for the WDT to expire. The time period is ((the multiplier specified by CpuWdtCountSel) * (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.</p> <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Multiplier</u></th> <th><u>BitsMultiplier</u></th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>4095</td> <td>6h63</td> </tr> <tr> <td>1h</td> <td>2047</td> <td>7h31</td> </tr> <tr> <td>2h</td> <td>1023</td> <td>8h8191</td> </tr> <tr> <td>3h</td> <td>511</td> <td>9h16383</td> </tr> <tr> <td>4h</td> <td>255</td> <td>Fh-AhReserved</td> </tr> <tr> <td>5h</td> <td>127</td> <td></td> </tr> </tbody> </table>	<u>Bits</u>	<u>Multiplier</u>	<u>BitsMultiplier</u>	0h	4095	6h63	1h	2047	7h31	2h	1023	8h8191	3h	511	9h16383	4h	255	Fh-AhReserved	5h	127	
<u>Bits</u>	<u>Multiplier</u>	<u>BitsMultiplier</u>																				
0h	4095	6h63																				
1h	2047	7h31																				
2h	1023	8h8191																				
3h	511	9h16383																				
4h	255	Fh-AhReserved																				
5h	127																					
2:1	<p><b>CpuWdtTimeBase: CPU watchdog timer time base.</b> Specifies the time base for the timeout period specified in CpuWdtCountSel.</p> <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>1.31 ms</td> </tr> <tr> <td>01b</td> <td>1.28 us</td> </tr> <tr> <td>10b</td> <td>Reserved</td> </tr> <tr> <td>11b</td> <td>Reserved</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Description</u>	00b	1.31 ms	01b	1.28 us	10b	Reserved	11b	Reserved											
<u>Bits</u>	<u>Description</u>																					
00b	1.31 ms																					
01b	1.28 us																					
10b	Reserved																					
11b	Reserved																					
0	<p><b>CpuWdtEn: CPU watchdog timer enable.</b> 1=The WDT is enabled.</p>																					

**MSRC001\_0080 Frequency Sensitivity Feedback Monitor Actual Count 0**

GP-write.

Bits	Description								
63:56	<p><b>ClassCode.</b> Read; GP-write. Value: 01h. Specifies the core frequency sensitivity class for this feedback monitor.</p> <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>00h</td> <td>Reserved</td> </tr> <tr> <td>01h</td> <td>Core frequency sensitivity.</td> </tr> <tr> <td>FFh-02h</td> <td>Reserved</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Description</u>	00h	Reserved	01h	Core frequency sensitivity.	FFh-02h	Reserved
<u>Bits</u>	<u>Description</u>								
00h	Reserved								
01h	Core frequency sensitivity.								
FFh-02h	Reserved								
55:0	<p><b>ActualCount.</b> Read; GP-write; updated-by-hardware. Reset: 0. Specifies the actual count of this feedback monitor. See <a href="#">CPUID Fn8000_0007_EAX[MaxWrapTime]</a>.</p>								

**MSRC001\_0081 Frequency Sensitivity Feedback Monitor Reference Count 0**

GP-write.

Bits	Description
63:56	<p><b>ClassCode.</b> Value: 01h. See: <a href="#">MSRC001_0080[ClassCode]</a>. Specifies the core frequency sensitivity class for this feedback monitor.</p>
55:0	<p><b>RefCount.</b> Read-only; updated-by-hardware. Reset: 0. Specifies the reference count of this feedback monitor. See <a href="#">CPUID Fn8000_0007_EAX[MaxWrapTime]</a>.</p>

**MSRC001\_0111 SMM Base Address (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 [2.4.8.2.5 \[SMM Save State\]](#)) and it is restored on returning

from SMM. The 16-bit CS (code segment) selector is loaded with SmmBase[19:4] on entering SMM. SmmBase[3:0] is required to be 0. 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 SmmBase with the new value.
- Normal WRMSR access to this register.

Bits	Description
63:32	Reserved.
31:0	<b>SmmBase</b> . IF <b>MSRC001_0015</b> [SmmLock] THEN Read-only; <b>Not-same-for-all</b> . ELSE Read-write; <b>Not-same-for-all</b> . ENDIF.

### **MSRC001\_0112 SMM TSeg Base Address (SMMAddr)**

Reset: 0000\_0000\_0000\_0000h.

See 2.4.8.2 [System Management Mode (SMM)] and 2.4.6.1 [Memory Access to the Physical Address Space]. See **MSRC001\_0113** for more information about the ASeg and TSeg address ranges.

Each CPU access, directed at CPUAddr, is determined to be in the TSeg range if the following is true:

$\text{CPUAddr}[39:17] \& \text{TSegMask}[39:17] == \text{TSegBase}[39:17] \& \text{TSegMask}[39:17]$ .

For example, if TSeg spans 256 KB and starts at the 1 MB 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\_0000 to 0013\_FFFFh.

Bits	Description
63:40	Reserved.
39:17	<b>TSegBase[39:17]: TSeg address range base</b> . IF <b>MSRC001_0015</b> [SmmLock] THEN Read-only ELSE Read-write ENDIF.
16:0	Reserved.

### **MSRC001\_0113 SMM TSeg Mask (SMMMMask)**

Reset: 0000\_0000\_0000\_0000h. See 2.4.8.2 [System Management Mode (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 controlled as follows:

- 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.
- See 2.4.6.1.1 [Determining Memory Type].

Bits	Description
63:40	Reserved.
39:17	<b>TSegMask[39:17]: TSeg address range mask.</b> IF MSRC001_0015[SmmLock] THEN Read-only ELSE Read-write ENDIF. See MSRC001_0112.
16:15	Reserved.
14:12	<b>TMTypeDram: TSeg address range memory type.</b> IF MSRC001_0015[SmmLock] THEN Read-only. ELSE Read-write. ENDIF. Specifies the memory type for SMM accesses to the TSeg range that are directed to DRAM. See: Table 192 [Valid Values for Memory Type Definition].
11	Reserved.
10:8	<b>AMTypeDram: ASeg Range Memory Type.</b> IF MSRC001_0015[SmmLock] THEN Read-only. ELSE Read-write. ENDIF. Specifies the memory type for SMM accesses to the ASeg range that are directed to DRAM. See: Table 192 [Valid Values for Memory Type Definition].
7:6	Reserved.
5	<b>TMTypeIoWc: non-SMM TSeg address range memory type.</b> IF MSRC001_0015[SmmLock] 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	<b>AMTypeIoWc: non-SMM ASeg address range memory type.</b> IF MSRC001_0015[SmmLock] 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	<b>TClose: send TSeg address range data accesses to MMIO.</b> Read-write. 1=When in SMM, direct data accesses in the TSeg address range to MMIO space. See AClose.
2	<b>AClose: send ASeg address range data accesses to MMIO.</b> Read-write. 1=When in SMM, direct data accesses in the ASeg address range to MMIO space.  [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	<b>TValid: enable TSeg SMM address range.</b> IF MSRC001_0015[SmmLock] THEN Read-only. ELSE Read-write. ENDIF. 1=The TSeg address range SMM enabled.
0	<b>AValid: enable ASeg SMM address range.</b> IF MSRC001_0015[SmmLock] THEN Read-only. ELSE Read-write. ENDIF. 1=The ASeg address range SMM enabled.

#### MSRC001\_0114 Virtual Machine Control (VM\_CR)

Bits	Description
63:32	Reserved.
31:5	MBZ.

4	<b>SvmeDisable: SVM disable.</b> See Lock for the access type of this field. Reset: 0. 1= <b>MSRC000_0080</b> [SVME] is MBZ. 0= <b>MSRC000_0080</b> [SVME] is read-write. Attempting to set this field when ( <b>MSRC000_0080</b> [SVME]==1) causes a #GP fault, regardless of the state of Lock. See the APM2 section titled “Enabling SVM” for software use of this field.
3	<b>Lock: SVM lock.</b> Read-only; write-1-only; cleared-by-hardware. Reset: 0. See <b>MSRC001_0118</b> [SvmLockKey] for the condition that causes hardware to clear this field. 1= <b>SvmeDisable</b> is read-only. 0= <b>SvmeDisable</b> is read-write.
2	<b>DisA20m: disable A20 masking.</b> Read-write; set-by-hardware. Reset: 0. 1=Disables A20 masking. This bit is set by hardware when the SKINIT instruction is executed.
1	<b>InterceptInit: intercept INIT.</b> Read-write; set-by-hardware. Reset: 0. This bit controls how INIT is delivered in host mode. This bit is set by hardware when the SKINIT instruction is executed. 0=INIT delivered normally. 1=INIT translated into a SX interrupt.
0	<b>DPD: debug port disable.</b> Read-write; set-by-hardware. Reset: 0. Set by hardware when the SKINIT instruction is executed. This bit controls if debug facilities such as JTAG and HDT have access to the processor state information. 1=HDT is disabled. 0=HDT may be enabled.

**MSRC001\_0115 IGNNE**

Bits	Description
63:32	Reserved.
31:1	MBZ.
0	<b>IGNNE: current IGNNE state.</b> Read-write. Reset: 0. This bit controls the current state of the processor internal IGNNE signal.

**MSRC001\_0116 SMM Control (SMM\_CTL)**

IF (**MSRC001\_0015**[SmmLock]) THEN GP-read-write. ELSE GP-read; write-only. ENDIF.

The bits in this register are processed in the order of: SmmEnter, SmiCycle, SmmDismiss, RsmCycle and SmmExit. However, only the following combination of bits may be set in a single write (all other combinations result in undefined behavior):

- SmmEnter and SmiCycle.
- SmmEnter and SmmDismiss.
- SmmEnter, SmiCycle and SmmDismiss.
- SmmExit and RsmCycle.

Software is responsible for ensuring that SmmEnter and SmmExit operations are properly matched and are not nested.

Bits	Description
63:5	MBZ.
4	<b>RsmCycle: send RSM special cycle.</b> 1=Send a RSM special cycle.
3	<b>SmmExit: exit SMM.</b> 1=Exit SMM.
2	<b>SmiCycle: send SMI special cycle.</b> 1=Send a SMI special cycle.
1	<b>SmmEnter: enter SMM.</b> 1=Enter SMM.
0	<b>SmmDismiss: clear SMI.</b> 1=Clear the SMI pending flag.

**MSRC001\_0117 Virtual Machine Host Save Physical Address (VM\_HSAVE\_PA)**

Bits	Description
63:40	MBZ.
39:12	<b>VM_HSAVE_PA: physical address of host save area.</b> Read-write. Reset: 0. This register contains the physical address of a 4-KB region where VMRUN saves host state and where vm-exit restores host state from. Writing this register causes a #GP if (FF_FFFF_Fh>=VM_HSAVE_PA>=FD_0000_0h).
11:0	MBZ.

**MSRC001\_0118 SVM Lock Key**

Reset: 0000\_0000\_0000\_0000h.

Bits	Description
63:0	<b>SvmLockKey: SVM lock key.</b> RAZ; write. Writes to this register when <a href="#">MSRC001_0114[Lock]</a> ==0 modify SvmLockKey. If (( <a href="#">MSRC001_0114[Lock]</a> ==1) && (SvmLockKey!=0) && (The write value==The value stored in SvmLockKey)) for a write to this register then hardware updates <a href="#">MSRC001_0114[Lock]</a> =0.

**MSRC001\_011A Local SMI Status**

Reset: 0000\_0000\_0000\_0000h. This registers returns the same information that is returned in [SMMFEC4 \[Local SMI Status\]](#) portion of the SMM save state. The information in this register is only updated when [MSRC001\\_0116\[SmmDismiss\]](#) is set by software.

Bits	Description
63:32	Reserved.
31:0	See <a href="#">SMMFEC4 [Local SMI Status]</a> .

**MSRC001\_0140 OS Visible Work-around MSR0 (OSVW\_ID\_Length)**

Reset: 0000\_0000\_0000\_0000h.

Bits	Description
63:16	Reserved.
15:0	<b>OSVWidLength: OS visible work-around ID length.</b> Read-write. See the Revision Guide for the definition of this field; see <a href="#">1.2 [Reference Documents]</a> .

**MSRC001\_0141 OS Visible Work-around MSR1 (OSVW Status)**

Reset: 0000\_0000\_0000\_0000h.

Bits	Description
63:0	<b>OsvwStatusBits: OS visible work-around status bits.</b> Read-write. See the Revision Guide for the definition of this field; see <a href="#">1.2 [Reference Documents]</a> .

**MSRC001\_023[6,4,2,0] L2I Performance Event Select (L2I\_PERF\_CTL[3:0])**

Per-L2. Reset: 0000\_0000\_0000\_0000h. See 2.6.1 [Performance Monitor Counters].

Table 224: Register Mapping for MSRC001\_023[6,4,2,0]

Register	Function
MSRC001_0230	Counter 0
MSRC001_0232	Counter 1
MSRC001_0234	Counter 2
MSRC001_0236	Counter 3

Bits	Description										
63:60	Reserved.										
59:56	<b>ThreadMask[3:0]</b> . Read-write. 0=Count events from the specified thread. 1=Mask count events from the specified thread. <table border="1"> <thead> <tr> <th>Bit</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>[0]</td> <td>Thread 0</td> </tr> <tr> <td>[1]</td> <td>Thread 1</td> </tr> <tr> <td>[2]</td> <td>Thread 2</td> </tr> <tr> <td>[3]</td> <td>Thread 3</td> </tr> </tbody> </table>	Bit	Description	[0]	Thread 0	[1]	Thread 1	[2]	Thread 2	[3]	Thread 3
Bit	Description										
[0]	Thread 0										
[1]	Thread 1										
[2]	Thread 2										
[3]	Thread 3										
55:52	Reserved.										
51:48	<b>BankMask[3:0]</b> . Read-write. 0=Count events from the specified bank. 1=Mask count events from the specified bank. <table border="1"> <thead> <tr> <th>Bit</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>[0]</td> <td>Bank 0</td> </tr> <tr> <td>[1]</td> <td>Bank 1</td> </tr> <tr> <td>[2]</td> <td>Bank 2</td> </tr> <tr> <td>[3]</td> <td>Bank 3</td> </tr> </tbody> </table>	Bit	Description	[0]	Bank 0	[1]	Bank 1	[2]	Bank 2	[3]	Bank 3
Bit	Description										
[0]	Bank 0										
[1]	Bank 1										
[2]	Bank 2										
[3]	Bank 3										
47:41	Reserved.										
40:37	<b>IntCoreSel: interrupt to core select</b> . Read-write. Specifies the logical core of the L2 complex to direct the interrupt.										
36	<b>IntCoreEn: interrupt to core enable</b> . Read-write. 1=Interrupt to a single core specified by IntCoreSel. 0=Interrupt to all cores.										
35:32	<b>EventSelect[11:8]: performance event select</b> . See: EventSelect[7:0].										
31:24	<b>CntMask: counter mask</b> . Read-write. Controls the number of events counted per clock cycle. <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00h</td> <td>The corresponding L2I_PERF_CTL[3:0] register increments by the number of events occurring in a clock cycle. Maximum number of events in one cycle is 4.</td> </tr> <tr> <td>03h-01h</td> <td>When Inv = 0, the corresponding L2I_PERF_CTL[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. When Inv = 1, the corresponding L2I_PERF_CTL[3:0] register increments by 1, if the number of events occurring in a clock cycle is less than CntMask value.</td> </tr> <tr> <td>FFh-04h</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Description	00h	The corresponding L2I_PERF_CTL[3:0] register increments by the number of events occurring in a clock cycle. Maximum number of events in one cycle is 4.	03h-01h	When Inv = 0, the corresponding L2I_PERF_CTL[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. When Inv = 1, the corresponding L2I_PERF_CTL[3:0] register increments by 1, if the number of events occurring in a clock cycle is less than CntMask value.	FFh-04h	Reserved		
Bits	Description										
00h	The corresponding L2I_PERF_CTL[3:0] register increments by the number of events occurring in a clock cycle. Maximum number of events in one cycle is 4.										
03h-01h	When Inv = 0, the corresponding L2I_PERF_CTL[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. When Inv = 1, the corresponding L2I_PERF_CTL[3:0] register increments by 1, if the number of events occurring in a clock cycle is less than CntMask value.										
FFh-04h	Reserved										
23	<b>Inv: invert counter mask</b> . Read-write. See CntMask.										
22	<b>En: enable performance counter</b> . Read-write. 1= Performance event counter is enabled.										

21	Reserved.
20	<b>Int: enable APIC interrupt.</b> Read-write. 1=APIC performance counter LVT interrupt is enabled to generate an interrupt via <a href="#">APIC340 [LVT Performance Monitor]</a> when the performance counter overflows.
19	Reserved.
18:16	Reserved.
15:8	<b>UnitMask: event qualification.</b> 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 <b>Unused</b> .
7:0	<b>EventSelect[7:0]: event select.</b> 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_CNT[3:0] register. The events are specified in section <a href="#">3.22 [Core Performance Counter Events]</a> . Some events are reserved; when a reserved event is selected, the results are undefined.

### **MSRC001\_023[7,5,3,1] L2I Performance Event Counter (L2I\_PERF\_CTR[3:0])**

Per-L2. See [MSRC001\\_023\[6,4,2,0\] \[L2I Performance Event Select \(L2I\\_PERF\\_CTL\[3:0\]\)\]](#).

Table 225: Register Mapping for [MSRC001\\_023\[7,5,3,1\]](#)

Register	Function
MSRC001_0231	Counter 0
MSRC001_0233	Counter 1
MSRC001_0235	Counter 2
MSRC001_0237	Counter 3

Bits	Description
63:48	RAZ.
47:0	<b>CTR: performance counter value.</b> Read-write. Reset: 0.

### **MSRC001\_024[6,4,2,0] Northbridge Performance Event Select (NB\_PERF\_CTL[3:0])**

Per-node. See [2.6.1 \[Performance Monitor Counters\]](#). [MSRC001\\_024\[6,4,2,0\]\[31:0\]](#) is an alias of [D18F5x\[70,60,50,40\]](#). [MSRC001\\_024\[6,4,2,0\]\[63:32\]](#) is an alias of [D18F5x\[74,64,54,44\]](#).

Table 226: Register Mapping for [MSRC001\\_024\[6,4,2,0\]](#)

Register	Function
MSRC001_0240	Counter 0
MSRC001_0242	Counter 1
MSRC001_0244	Counter 2
MSRC001_0246	Counter 3

Bits	Description
63:36	Reserved.
35:32	<b>EventSelect[11:8]: performance event select.</b> Read-write. Reset: 0. See EventSelect[7:0].
31:23	Reserved.
22	<b>En: enable performance counter.</b> Read-write. Reset: 0. 1= Performance event counter is enabled.
21	Reserved.
20	<b>Int: enable APIC interrupt.</b> Read-write. Reset: 0. 1=APIC performance counter LVT interrupt is enabled to generate an interrupt via <a href="#">APIC340 [LVT Performance Monitor]</a> to all local APIC's on this node when the performance counter overflows.
19	Reserved.
18:16	Reserved.
15:8	<b>UnitMask: event qualification.</b> Read-write. Reset: 0. 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 <a href="#">Unused</a> .
7:0	<b>EventSelect[7:0]: event select.</b> Read-write. Reset: 0. 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 NB_PERF_CTR[3:0] register. The events are specified in <a href="#">3.24 [NB Performance Counter Events]</a> . Some events are reserved; when a reserved event is selected, the results are undefined.

### **MSRC001\_024[7,5,3,1] Northbridge Performance Event Counter (NB\_PERF\_CTR[3:0])**

Per-node. See [MSRC001\\_024\[6,4,2,0\] \[Northbridge Performance Event Select \(NB\\_PERF\\_CTL\[3:0\]\)\]](#). [MSRC001\\_024\[7,5,3,1\]\[31:0\]](#) is an alias of [D18F5x\[78,68,58,48\]](#). [MSRC001\\_024\[7,5,3,1\]\[63:32\]](#) is an alias of [D18F5x\[7C,6C,5C,4C\]](#).

Table 227: Register Mapping for [MSRC001\\_024\[7,5,3,1\]](#)

Register	Function
MSRC001_0241	Counter 0
MSRC001_0243	Counter 1
MSRC001_0245	Counter 2
MSRC001_0247	Counter 3

Bits	Description
63:48	RAZ.
47:32	<b>CTR[47:32]: performance counter value[47:32].</b> See: CTR[31:0].
31:0	<b>CTR[31:0]: performance counter value[31:0].</b> Read-write. Reset: 0. CTR[47:0] = {CTR[47:32], CTR[31:0]}. Returns the current value of the event counter.



**3.21 MSRs - MSRC001\_1xxx****MSRC001\_1002 CPUID Features for CPUID Fn0000\_0007\_E[B,A]X\_x0**

Read-write. Reset: {CPUID Fn0000\_0007\_EAX\_x0, CPUID Fn0000\_0007\_EBX\_x0}.

MSRC001\_1002[63:32] provides control over values read from CPUID Fn0000\_0007\_EAX\_x0;

MSRC001\_1002[31:0] provides control over values read from CPUID Fn0000\_0007\_EBX\_x0.

Bits	Description
63:9	Reserved.
8	Reserved.
7	Reserved.
6	Reserved.
5	Reserved.
4	Reserved.
3	<b>BMI1.</b>
2:1	Reserved.
0	Reserved.

**MSRC001\_1004 CPUID Features (Features)**

Read-write. Reset: {CPUID Fn0000\_0001\_ECX, CPUID Fn0000\_0001\_EDX}. MSRC001\_1004[63:32] provides control over values read from CPUID Fn0000\_0001\_ECX; MSRC001\_1004[31:0] provides control over values read from CPUID Fn0000\_0001\_EDX.

Bits	Description
63	Reserved.
62	<b>RDRAND.</b>
61	<b>F16C.</b>
60	<b>AVX.</b>
59	<b>OSXSAVE.</b> Modifies CPUID Fn0000_0001_ECX[OSXSAVE] only if CR4[OSXSAVE].
58	<b>XSAVE.</b>
57	<b>AES.</b> Modifies CPUID Fn0000_0001_ECX[AES] only if the reset value is 1 .
56	Reserved.
55	<b>POPCNT.</b>
54	<b>MOVBE.</b>
53	<b>x2APIC.</b>
52	<b>SSE42.</b>
51	<b>SSE41.</b>

50:46	Reserved.
45	<b>CMPXCHG16B.</b>
44:42	Reserved.
41	<b>SSSE3.</b>
40:36	Reserved.
35	<b>Monitor.</b> Modifies <a href="#">CUID Fn0000_0001_ECX[Monitor]</a> only if <a href="#">~MSRC001_0015[MonMwaitDis]</a> .
34	Reserved.
33	<b>PCLMULQDQ.</b> Modifies <a href="#">CUID Fn0000_0001_ECX[PCLMULQDQ]</a> only if the reset value is 1
32	<b>SSE3.</b>
31:29	Reserved.
28	<b>HTT.</b>
27	Reserved.
26	<b>SSE2.</b>
25	<b>SSE.</b>
24	<b>FXSR.</b>
23	<b>MMX.</b>
22:20	Reserved.
19	<b>CLFSH.</b>
18	Reserved.
17	<b>PSE36.</b>
16	<b>PAT.</b>
15	<b>CMOV.</b>
14	<b>MCA.</b>
13	<b>PGE.</b>
12	<b>MTRR.</b>
11	<b>SysEnterSysExit.</b>
10	Reserved.
9	<b>APIC.</b> Modifies <a href="#">CUID Fn0000_0001_EDX[APIC]</a> only if <a href="#">MSR0000_001B[ApicEn]</a> .
8	<b>CMPXCHG8B.</b>
7	<b>MCE.</b>
6	<b>PAE.</b>
5	<b>MSR.</b>
4	<b>TSC.</b>
3	<b>PSE.</b>
2	<b>DE.</b>
1	<b>VME.</b>
0	<b>FPU.</b>

**MSRC001\_1005 Extended CPUID Features (ExtFeatures)**

Read-write. Reset: {CPUID Fn8000\_0001\_ECX, CPUID Fn8000\_0001\_EDX}. MSRC001\_1005[63:32] provides control over values read from CPUID Fn8000\_0001\_ECX; MSRC001\_1005[31:0] provides control over values read from CPUID Fn8000\_0001\_EDX.

Bits	Description
63:61	Reserved.
60	<b>PerfCtrExtL2I.</b>
59	<b>PerfTsc.</b>
58	<b>DataBreakpointExtension.</b>
57	Reserved.
56	<b>PerfCtrExtNB.</b>
55	<b>PerfCtrExtCore.</b>
54	<b>TopologyExtensions.</b>
53	<b>TBM.</b>
52	Reserved.
51	<b>NodeId.</b>
50	Reserved.
49	Reserved.
48	<b>FMA4.</b>
47	<b>LWP.</b>
46	Reserved.
45	<b>WDT.</b>
44	<b>SKINIT.</b>
43	<b>XOP.</b>
42	<b>IBS.</b>
41	<b>OSVW.</b>
40	<b>3DNowPrefetch.</b>
39	<b>MisAlignSse.</b>
38	<b>SSE4A.</b>
37	<b>ABM.</b>
36	<b>AltMovCr8.</b>
35	<b>ExtApicSpace.</b>
34	<b>SVM.</b> Modifies CPUID Fn8000_0001_ECX[SVM] only if D18F3xE8[SvmCapable].
33	<b>CmpLegacy.</b>
32	<b>LahfSahf.</b>
31	<b>3DNow.</b>
30	<b>3DNowExt.</b>
29	<b>LM.</b> Read-write.

28	Reserved.
27	<b>RDTS CP.</b>
26	<b>Page1GB.</b>
25	<b>FFXS R.</b>
24	<b>FXS R.</b>
23	<b>MMX.</b>
22	<b>MmxExt.</b>
21	Reserved.
20	<b>NX.</b>
19:18	Reserved.
17	<b>PSE36.</b>
16	<b>PAT.</b>
15	<b>CMOV.</b>
14	<b>MCA.</b>
13	<b>PGE.</b>
12	<b>MTRR.</b>
11	<b>SysCallSysRet.</b>
10	Reserved.
9	<b>APIC.</b>
8	<b>CMPXCHG8B.</b>
7	<b>MCE.</b>
6	<b>PAE.</b>
5	<b>MSR.</b>
4	<b>TSC.</b>
3	<b>PSE.</b>
2	<b>DE.</b>
1	<b>VME.</b>
0	<b>FPU.</b>

### MSRC001\_101[B:9] Address Mask For DR[3:1] Breakpoints

Reset: 0000\_0000\_0000\_0000h. Support indicated by CPUID Fn8000\_0001\_ECX[DataBreakpointExtension]. See [MSRC001\\_1027](#).

Table 228: [Register Mapping](#) for MSRC001\_101[B:9]

Register	Function
MSRC001_1019	DR1_ADDR_MASK
MSRC001_101A	DR2_ADDR_MASK
MSRC001_101B	DR3_ADDR_MASK

Table 229: Field Mapping for MSRC001\_101[B:9]

Register	Bits
	31:0
MSRC001_1019	DR1
MSRC001_101A	DR2
MSRC001_101B	DR3

Bits	Description
63:32	Reserved.
31:0	<b>AddrMask: mask for DR linear address data breakpoint.</b> Read-write. This field qualifies the DR linear address data breakpoint, allowing the DR[3:1] data breakpoint on a range of addresses in memory. The mask bits are active high; 0=Include bit into address compare; 1=Exclude bit into address compare. AddrMask is always used, and it can be used in conjunction with any debug function that uses DR[3:1]. The legacy DR breakpoint function is provided by AddrMask[31:0]==0000_0000h).

**MSRC001\_1020 Load-Store Configuration (LS\_CFG)**

Bits	Description
63:53	Reserved.
52	Reserved.
51	Reserved.
50:45	Reserved.
44:27	Reserved.
26	<b>DisHitCurPageOpt.</b> Read-write. Reset: 0. BIOS: 1. 1=Disable current table-walk page hit optimization.
25	Reserved.
24	Reserved.
23:0	Reserved.

**MSRC001\_1021 Instruction Cache Configuration (IC\_CFG)**

Bits	Description
63:27	Reserved.

26	<b>DIS_WIDEREAD_PWR_SAVE.</b> Read-write; <b>Same-for-all.</b> Reset: 0. BIOS: 1. 1=Disable wide read power mgmt feature for read accesses to IC data array. See 2.3.4 [Instruction Cache Configuration Register Usage Requirements].
25:10	Reserved.
9	<b>DisSpecTlbRld.</b> Read-write; <b>Same-for-all.</b> Reset: 0. BIOS: See 2.3.3 [Using L2 Cache as General Storage During Boot]. 1=Disable speculative TLB reloads. See 2.3.4 [Instruction Cache Configuration Register Usage Requirements].
8:0	Reserved.

### MSRC001\_1022 Data Cache Configuration (DC\_CFG)

Bits	Description
63:14	Reserved.
13	<b>DisHwPf.</b> Read-write. Reset: 0. BIOS: See 2.3.3. 1=Disable hardware prefetches.
12:5	Reserved.
4	<b>DisSpecTlbWalk.</b> Read-write. Reset: 0. BIOS: See 2.3.3. 1=Disable speculative table-walks.
3:0	Reserved.

### MSRC001\_1023 Bus Unit Configuration (BU\_CFG)

Bits	Description
63:0	Reserved.

### MSRC001\_1027 Address Mask For DR0 Breakpoints (DR0\_ADDR\_MASK)

Reset: 0000\_0000\_0000\_0000h. Support for AddrMaskDR0[31:12] is indicated by **CPUID Fn8000\_0001\_ECX[DataBreakpointExtension]**. See **MSRC001\_101[B:9]**.

Bits	Description
63:32	Reserved.
31:0	<b>AddrMaskDR0: mask for DR0 linear address data breakpoint.</b> Read-write. This field qualifies the DR0 linear address data breakpoint, allowing the DR0 data breakpoint on a range of addresses in memory. The mask bits are active high; 0=Include bit into address compare; 1=Exclude bit into address compare. AddrMaskDR0 is always used, and it can be used in conjunction with any debug function that uses DR0. AddrMaskDR0[31:12] is only valid for data breakpoints. The legacy DR0 breakpoint function is provided by AddrMaskDR0[31:0]==0000_0000h).

### MSRC001\_1028 Floating Point Configuration (FP\_CFG)

Bits	Description
63:0	Reserved.

**MSRC001\_102A Bus Unit Configuration 2 (BU\_CFG2)**

Bits	Description
63:16	Reserved.
15	<b>CILinesToL2Dis</b> . Read-write. Reset: 1. BIOS: See 2.3.3 [Using L2 Cache as General Storage During Boot]. 1=Clean victims and clean forced victims are not sent to the L2.
14:0	Reserved.

**MSRC001\_1030 IBS Fetch Control (IbsFetchCtl)**

Reset: 0000\_0000\_0000\_0000h. See 2.6.2 [Instruction Based Sampling (IBS)].

The IBS fetch sampling engine is described as follows:

- The periodic fetch counter is an internal 20-bit counter:
  - The periodic fetch counter [19:4] is set to IbsFetchCnt[19:4] and the periodic fetch counter [3:0] is set according to IbsRandEn when IbsFetchEn is changed from 0 to 1.
  - It increments for every fetch cycle that completes when IbsFetchEn=1 and IbsFetchVal=0.
    - The periodic fetch counter is undefined when IbsFetchEn=0 or IbsFetchVal=1.
  - When IbsFetchCnt[19:4] is read it returns the current value of the periodic fetch counter [19:4].
- When the periodic fetch counter reaches {IbsFetchMaxCnt[19:4],0h} and the selected instruction fetch completes or is aborted:
  - IbsFetchVal is set to 1.
    - Drivers can't assume that IbsFetchCnt[19:4] is 0 when IbsFetchVal==1.
  - The status of the operation is written to the IBS fetch registers (this register, MSRC001\_1031 and MSRC001\_1032).
  - 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.

Bits	Description										
63:59	Reserved.										
58	Reserved.										
57	<b>IbsRandEn: random instruction fetch tagging enable</b> . Read-write. 1=Bits[3:0] of the fetch counter are randomized when IbsFetchEn is set to start the fetch counter. 0=Bits[3:0] of the fetch counter are set to 0h when IbsFetchEn is set to start the fetch counter.										
56	<b>IbsL2TlbMiss: instruction cache L2TLB miss</b> . Read-write; set-by-hardware. 1=The instruction fetch missed in the L2 TLB.										
55	<b>IbsL1TlbMiss: instruction cache L1TLB miss</b> . Read-write; set-by-hardware. 1=The instruction fetch missed in the L1 TLB.										
54:53	<b>IbsL1TlbPgSz: instruction cache L1TLB page size</b> . Read-write; updated-by-hardware. Indicates the page size of the translation in the L1 TLB. This field is only valid if IbsPhyAddrValid==1. <table border="1" data-bbox="267 1732 665 1900"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>4 KB</td> </tr> <tr> <td>01b</td> <td>2 MB</td> </tr> <tr> <td>10b</td> <td>Reserved</td> </tr> <tr> <td>11b</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Description	00b	4 KB	01b	2 MB	10b	Reserved	11b	Reserved
Bits	Description										
00b	4 KB										
01b	2 MB										
10b	Reserved										
11b	Reserved										

52	<b>IbsPhyAddrValid: instruction fetch physical address valid.</b> Read-write; set-by-hardware. 1=The physical address in <a href="#">MSRC001_1032</a> and the IbsL1TlbPgSz field are valid for the instruction fetch.
51	<b>IbsIcMiss: instruction cache miss.</b> Read-write; set-by-hardware. 1=The instruction fetch missed in the instruction cache.
50	<b>IbsFetchComp: instruction fetch complete.</b> Read-write; set-by-hardware. 1=The instruction fetch completed and the data is available for use by the instruction decoder.
49	<b>IbsFetchVal: instruction fetch valid.</b> Read-write; set-by-hardware. 1=New instruction fetch data available. When this bit is set, the fetch counter stops counting and an interrupt is generated as specified by <a href="#">MSRC001_103A</a> . This bit must be cleared for the fetch counter to start counting. When clearing this bit, software can write 0000h to IbsFetchCnt[19:4] to start the fetch counter at IbsFetchMaxCnt[19:4].
48	<b>IbsFetchEn: instruction fetch enable.</b> Read-write. 1=Instruction fetch sampling is enabled.
47:32	<b>IbsFetchLat: instruction fetch latency.</b> Read-write; set-by-hardware. 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	<b>IbsFetchCnt[19:4].</b> Read-write; updated-by-hardware. Provides read/write access to bits[19:4] of the periodic fetch counter. Programming this field to a value greater than or equal to IbsFetchMaxCnt[19:4] results in undefined behavior.
15:0	<b>IbsFetchMaxCnt[19:4].</b> Read-write. Specifies bits[19:4] of the maximum count value of the periodic fetch counter. Programming this field to 0000h and setting IbsFetchEn results in undefined behavior. Bits[3:0] of the maximum count are always 0000b.

### MSRC001\_1031 IBS Fetch Linear Address (IbsFetchLinAd)

Reset: 0000\_0000\_0000\_0000h.

Bits	Description
63:0	<b>IbsFetchLinAd: instruction fetch linear address.</b> Read-write; updated-by-hardware. Provides the linear address in canonical form for the tagged instruction fetch.

### MSRC001\_1032 IBS Fetch Physical Address (IbsFetchPhysAd)

Reset: 0000\_0000\_0000\_0000h.

Bits	Description
63:40	Reserved.
39:0	<b>IbsFetchPhysAd: instruction fetch physical address.</b> Read-write; updated-by-hardware. Provides the physical address 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 <a href="#">MSRC001_1030</a> [IbsPhyAddrValid] is asserted.

### MSRC001\_1033 IBS Execution Control (IbsOpCtl)

Reset: 0000\_0000\_0000\_0000h. See [2.6.2 \[Instruction Based Sampling \(IBS\)\]](#).

The IBS execution sampling engine is described as follows for IbsOpCntCtl==1. If IbsOpCntCtl==1n then ref-



ences to “periodic op counter” mean “periodic cycle counter”.

- The periodic op counter is an internal 27-bit counter:
  - It is set to IbsOpCurCnt[26:0] when IbsOpEn is changed from 0 to 1.
  - It increments every dispatched op when IbsOpEn=1 and IbsOpVal=0.
    - The periodic op counter is undefined when IbsOpEn=0 or IbsOpVal=1.
  - When IbsOpCurCnt[26:0] is read then it returns the current value of the periodic micro-op counter [26:0].
- When the periodic micro-op counter reaches IbsOpMaxCnt:
  - The next dispatched micro-op is tagged if IbsOpCntCtl==1. A valid op in the next dispatched line is tagged if IbsOpCntCtl==0. See IbsOpCntCtl.
  - The periodic micro-op counter [26:7]=0; [6:0] is randomized by hardware.
- The periodic micro-op counter is not modified when a tagged micro-op is flushed.
- When a tagged micro-op is retired:
  - IbsOpVal is set to 1.
    - Drivers can’t assume that IbsOpCurCnt is 0 when IbsOpVal==1.
  - 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](#)).
  - 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.

Bits	Description						
63:59	Reserved.						
58:32	<b>IbsOpCurCnt[26:0]: periodic op counter current count.</b> Read-write; updated-by-hardware. Returns the current value of the periodic op counter.						
31:27	Reserved.						
26:20	<b>IbsOpMaxCnt[26:20]: periodic op counter maximum count.</b> Read-write. See IbsOpMaxCnt[19:4].						
19	<b>IbsOpCntCtl: periodic op counter count control.</b> Read-write. 1=Count dispatched <a href="#">Micro-ops</a> ; when a roll-over occurs, the counter is preloaded with a pseudorandom 7 bit value between 1 and 127. 0=Count clock cycles; a 1-of-4 round robin counter selects an op in the next dispatch line; if the op pointed to by the round robin counter is invalid, then the next younger valid op is selected.						
18	<b>IbsOpVal: micro-op sample valid.</b> Read-write; set-by-hardware. 1=New instruction execution data available; the periodic op counter is disabled from counting. An interrupt may be generated when this bit is set as specified by <a href="#">MSRC001_103A</a> [LvtOffset].						
17	<b>IbsOpEn: micro-op sampling enable.</b> Read-write. 1=Instruction execution sampling enabled.						
16	Reserved.						
15:0	<b>IbsOpMaxCnt[19:4]: periodic op counter maximum count.</b> Read-write. IbsOpMaxCnt[26:0] = {IbsOpMaxCnt[26:20], IbsOpMaxCnt[19:4], 0000b}. Specifies maximum count value of the periodic op counter. Bits [3:0] of the maximum count are always 0000b. <table border="0" style="margin-left: 20px;"> <tr> <td style="padding-right: 20px;"><u>Bits</u></td> <td><u>Description</u></td> </tr> <tr> <td>001Fh-0000h</td> <td>Reserved</td> </tr> <tr> <td>FFFFh-0020h</td> <td>&lt;IbsOpMaxCnt[19:4]*16&gt; <a href="#">Micro-ops</a></td> </tr> </table>	<u>Bits</u>	<u>Description</u>	001Fh-0000h	Reserved	FFFFh-0020h	<IbsOpMaxCnt[19:4]*16> <a href="#">Micro-ops</a>
<u>Bits</u>	<u>Description</u>						
001Fh-0000h	Reserved						
FFFFh-0020h	<IbsOpMaxCnt[19:4]*16> <a href="#">Micro-ops</a>						

**MSRC001\_1034 IBS Op Logical Address (IbsOpRip)**

Reset: 0000\_0000\_0000\_0000h.

Bits	Description
63:0	<b>IbsOpRip: micro-op linear address.</b> Read-write; updated-by-hardware. Linear address in canonical form for the instruction that contains the tagged micro-op.

**MSRC001\_1035 IBS Op Data (IbsOpData)**

Bits	Description
63:41	Reserved.
40	<b>IbsOpMicrocode.</b> Value: 0. 1=Tagged operation from microcode.
39	<b>IbsOpBrnFuse: fused branch micro-op.</b> Value: 0. 1=Tagged operation was a fused branch micro-op. Support indicated by <a href="#">CPUID Fn8000_001B_EAX[OpBrnFuse]</a> .
38	<b>IbsRipInvalid: RIP is invalid.</b> Read-write; updated-by-hardware. Reset: 0. 1=Tagged operation RIP is invalid. Support indicated by <a href="#">CPUID Fn8000_001B_EAX[RipInvalidChk]</a> .
37	<b>IbsOpBrnRet: branch micro-op retired.</b> Read-write; updated-by-hardware. Reset: 0. 1=Tagged operation was a branch micro-op that retired.
36	<b>IbsOpBrnMisp: mispredicted branch micro-op.</b> Read-write; updated-by-hardware. Reset: 0. 1=Tagged operation was a branch micro-op that was mispredicted. Qualified by <code>IbsOpBrnRet==1</code> .
35	<b>IbsOpBrnTaken: taken branch micro-op.</b> Read-write; updated-by-hardware. Reset: 0. 1=Tagged operation was a branch micro-op that was taken. Qualified by <code>IbsOpBrnRet==1</code> .
34	<b>IbsOpReturn: return micro-op.</b> Read-write; updated-by-hardware. Reset: 0. 1=Tagged operation was return micro-op. Qualified by <code>(IbsOpBrnRet==1)</code> .
33:32	Reserved.
31:16	<b>IbsTagToRetCtr: micro-op tag to retire count.</b> Read-write; updated-by-hardware. Reset: 0. 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 <code>IbsCompToRetCtr</code> when the tagged micro-op is a NOP.
15:0	<b>IbsCompToRetCtr: micro-op completion to retire count.</b> Read-write; updated-by-hardware. Reset: 0. 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 (IbsOpData2)**

Reset: 0000\_0000h. 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	Reserved.																		
4	Reserved.																		
3	Reserved.																		
2:0	<p><b>NbIbsReqSrc: northbridge IBS request data source.</b> Read-write.</p> <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>No valid status</td> </tr> <tr> <td>1h</td> <td>Reserved</td> </tr> <tr> <td>2h</td> <td>Cache: data returned from another compute-unit cache.</td> </tr> <tr> <td>3h</td> <td>DRAM: data returned from DRAM</td> </tr> <tr> <td>4h</td> <td>Reserved for remote cache</td> </tr> <tr> <td>5h</td> <td>Reserved</td> </tr> <tr> <td>6h</td> <td>Reserved</td> </tr> <tr> <td>7h</td> <td>Other: data returned from MMIO/Config/PCI/APIC</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Description</u>	0h	No valid status	1h	Reserved	2h	Cache: data returned from another compute-unit cache.	3h	DRAM: data returned from DRAM	4h	Reserved for remote cache	5h	Reserved	6h	Reserved	7h	Other: data returned from MMIO/Config/PCI/APIC
<u>Bits</u>	<u>Description</u>																		
0h	No valid status																		
1h	Reserved																		
2h	Cache: data returned from another compute-unit cache.																		
3h	DRAM: data returned from DRAM																		
4h	Reserved for remote cache																		
5h	Reserved																		
6h	Reserved																		
7h	Other: data returned from MMIO/Config/PCI/APIC																		

### MSRC001\_1037 IBS Op Data 3 (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	<p><b>IbsTlbRefillLat: L1 DTLB refill latency.</b> Read-only. Value: 0. The number of cycles from when a L1 DTLB refill is triggered by a tagged op to when the L1 DTLB fill has been completed.</p>
47:32	<p><b>IbsDcMissLat: data cache miss latency.</b> Read-write; updated-by-hardware. 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 or prefetch instructions.</p>
31:26	<p><b>IbsOpDcMissOpenMemReqs: outstanding memory requests on DC fill.</b> Read-only. Value: 0. The number of allocated, valid DC MABs when the MAB corresponding to a tagged DC miss op is deallocated. Includes the MAB allocated by the sampled op. 00000b=No information provided.</p>

25:22	<p><b>IbsOpMemWidth: load/store size in bytes.</b> Read-only. Value: 0. Report the number of bytes the load or store is attempting to access.</p> <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>No information provided.</td> </tr> <tr> <td>1h</td> <td>Byte</td> </tr> <tr> <td>2h</td> <td>Word</td> </tr> <tr> <td>3h</td> <td>DW</td> </tr> <tr> <td>4h</td> <td>QW</td> </tr> <tr> <td>5h</td> <td>OW</td> </tr> <tr> <td>Fh-6h</td> <td>Reserved</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Description</u>	0h	No information provided.	1h	Byte	2h	Word	3h	DW	4h	QW	5h	OW	Fh-6h	Reserved
<u>Bits</u>	<u>Description</u>																
0h	No information provided.																
1h	Byte																
2h	Word																
3h	DW																
4h	QW																
5h	OW																
Fh-6h	Reserved																
21	<p><b>IbsSwPf: software prefetch.</b> Read-only. Value: 0. 1=The op is a software prefetch.</p>																
20	<p><b>IbsL2Miss: L2 cache miss for the sampled operation.</b> Read-only. Value: 0. 1=The operation missed in the L2, regardless of whether the op initiated the request to the L2.</p>																
19	<p><b>IbsDcL2TlbHit1G: data cache L2TLB hit in 1G page.</b> Read-only. Value: 0. 1=The physical address for the tagged load or store operation was present in a 1G page table entry in the data cache L2TLB.</p>																
18	<p><b>IbsDcPhyAddrValid: data cache physical address valid.</b> Read-write; updated-by-hardware. 1=The physical address in <a href="#">MSRC001_1039</a> is valid for the load or store operation.</p>																
17	<p><b>IbsDcLinAddrValid: data cache linear address valid.</b> Read-write; updated-by-hardware. 1=The linear address in <a href="#">MSRC001_1038</a> is valid for the load or store operation.</p>																
16	<p><b>DcMissNoMabAlloc: DC miss with no MAB allocated.</b> Read-write; updated-by-hardware. 1=The tagged load or store operation hit on an already allocated MAB.</p>																
15	<p><b>IbsDcLockedOp: locked operation.</b> Read-write; updated-by-hardware. 1=Tagged load or store operation is a locked operation.</p>																
14	<p><b>IbsDcUcMemAcc: UC memory access.</b> Read-write; updated-by-hardware. 1=Tagged load or store operation accessed uncacheable memory.</p>																
13	<p><b>IbsDcWcMemAcc: WC memory access.</b> Read-write; updated-by-hardware. 1=Tagged load or store operation accessed write combining memory.</p>																
12	Reserved.																
11	<p><b>IbsDcStToLdFwd: data forwarded from store to load operation.</b> Read-write; updated-by-hardware. 1=Data for tagged load operation was forwarded from a store operation. If this bit is set and <a href="#">IbsDcStToLdCan</a>=1, then the data for the load operation forwarded from a store operation but the data was not forwarded immediately.</p>																
10	Reserved.																
9	Reserved.																
8	<p><b>IbsDcMisAcc: misaligned access.</b> Read-write; updated-by-hardware. 1=The tagged load or store operation crosses a 128 bit address boundary.</p>																

7	<b>IbsDcMiss: data cache miss.</b> Read-write; updated-by-hardware. 1=The cache line used by the tagged load or store was not present in the data cache.
6	<b>IbsDcL2tlbHit2M: data cache L2TLB hit in 2M page.</b> Read-write; updated-by-hardware. 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	Reserved.
4	<b>IbsDcL1TlbHit2M: data cache L1TLB hit in 2M page.</b> Read-write; updated-by-hardware. 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	<b>IbsDcL2TlbMiss: data cache L2TLB miss.</b> Read-write; updated-by-hardware. 1=The physical address for the tagged load or store operation was not present in the data cache L2TLB.
2	<b>IbsDcL1tlbMiss: data cache L1TLB miss.</b> Read-write; updated-by-hardware. 1=The physical address for the tagged load or store operation was not present in the data cache L1TLB.
1	<b>IbsStOp: store op.</b> Read-write; updated-by-hardware. 1=Tagged operation is a store operation.
0	<b>IbsLdOp: load op.</b> Read-write; updated-by-hardware. 1=Tagged operation is a load operation.

#### MSRC001\_1038 IBS DC Linear Address (IbsDcLinAd)

Reset: 0000\_0000\_0000\_0000h.

Bits	Description
63:0	<b>IbsDcLinAd.</b> Read-write; updated-by-hardware. Provides the linear address in canonical form for the tagged load or store operation. This field contains valid data only if <a href="#">MSRC001_1037</a> [IbsDcLinAddrValid] is asserted.

#### MSRC001\_1039 IBS DC Physical Address (IbsDcPhysAd)

Bits	Description
63:40	RAZ.
39:0	<b>IbsDcPhysAd: load or store physical address.</b> Read-write; updated-by-hardware. Reset: 0. 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 <a href="#">MSRC001_1037</a> [IbsDcPhyAddrValid] is asserted.

#### MSRC001\_103A IBS Control

GP-write.

Bits	Description
63:32	Reserved.
31:9	Reserved.
8	<b>LvtOffsetVal: local vector table offset valid.</b> <a href="#">MSRC001_103A</a> [LvtOffsetVal] is an alias of <a href="#">D18F3x1CC</a> [LvtOffsetVal].

7:4	Reserved.
3:0	<b>LvtOffset: local vector table offset.</b> <a href="#">MSRC001_103A[LvtOffset]</a> is an alias of <a href="#">D18F3x1CC[LvtOffset]</a> .

**MSRC001\_103B IBS Branch Target Address (BP\_IBSTGT\_RIP)**

Reset: 0000\_0000\_0000\_0000h. Support for this register indicated by [CPUID Fn8000\\_001B\\_EAX\[BrnTrgt\]](#).

Bits	Description
63:0	<b>IbsBrTarget.</b> Read-write; updated-by-hardware. The logical address in canonical form for the branch target. Contains a valid target if non-0. Qualified by <a href="#">MSRC001_1035[IbsOpBrnRet]</a> =1.

**MSRC001\_1090 Processor Feedback Constants 0**

Read-write.

Bits	Description
63:32	Reserved.
31:16	Reserved.
15:8	<b>RefCountScale.</b> Reset: Product-specific. BIOS: 64h.
7:0	<b>ActualCountScale.</b> Reset: Product-specific. BIOS: A5h.

**MSRC001\_10A0 L2I Configuration (L2I\_CFG)**

Per-L2.

Bits	Description				
63:57	Reserved.				
56:45	<b>L2ScrubberInterval: L2 background scrubber interval [11:0].</b> Read-write. Reset: 0. BIOS: 100h. If ( <a href="#">L2ScrubberDis</a> ==0), the background error scrubber will wake up and perform one scrub every (N+1)*64 clock CClks. <table border="0"> <tr> <td><u>Bits</u></td> <td><u>Scrubber Interval</u></td> </tr> <tr> <td>FFh-00h</td> <td>&lt;<a href="#">L2ScrubberInterval</a>&gt;+1) * 64 CClks</td> </tr> </table>	<u>Bits</u>	<u>Scrubber Interval</u>	FFh-00h	< <a href="#">L2ScrubberInterval</a> >+1) * 64 CClks
<u>Bits</u>	<u>Scrubber Interval</u>				
FFh-00h	< <a href="#">L2ScrubberInterval</a> >+1) * 64 CClks				
44	<b>PbDisObeyThrottleNb: probes disobey NB throttling.</b> Read-write. Reset: 0. BIOS: 1. 1=Probe responses do not obey the <a href="#">ThrottleNbInterface</a> field, but can be sent to the NB on any valid CClk.				

43:40	<p><b>ThrottleNbInterface: NB Interface Request Throttling [3:0]</b>. Read-write. Reset: Fh. BIOS: <a href="#">NumOfCompUnitsOnNode-1</a>. If (PbDisObeysThrottleNb==1), Specifies how many clocks the CU needs to wait before sending the next packet of information to the NB. This applies to the CU-&gt;NB request interface and the CU-&gt;NB probe response interface.</p> <p>This field must be programmed to a value greater than or equal to the number of L2 complexes in the node that have at least one enabled core minus 1. See <a href="#">2.4.4 [Processor Cores and Downcoring]</a>. If (PbDisObeysThrottleNb==0), probe responses must also obey the same throttling rule.</p> <table border="0"> <tr> <td style="text-align: center;"><u>Bits</u></td> <td style="text-align: center;"><u>Throttling Interval</u></td> </tr> <tr> <td style="text-align: center;">Fh-0h</td> <td style="text-align: center;">&lt;ThrottleNbInterface&gt;+1 CClks</td> </tr> </table>	<u>Bits</u>	<u>Throttling Interval</u>	Fh-0h	<ThrottleNbInterface>+1 CClks
<u>Bits</u>	<u>Throttling Interval</u>				
Fh-0h	<ThrottleNbInterface>+1 CClks				
39:22	Reserved.				
21	<b>L2ScrubberDis: L2 Scrubber Disable</b> . Read-write. Reset: 0. 1=Do not search for and repair single bit errors in the background.				
20	<b>L2RinserDis: L2 rinser disable</b> . Read-write. Reset: 0. BIOS: 1. 1=Disable background copy of dirty lines to the NB. See <a href="#">L2RinserInterval</a> .				
19	<b>McaToMstCoreEn: route machine check events to master Core</b> . Read-write. Reset: 0. BIOS: 1. 1=Errors are logged and reported to the L2 complex Master Core, which is the lowest physically numbered core in the L2 complex that is not fused off. 0=Errors are logged and reported to the core whose behavior triggered the error, if that can be determined. If it cannot be determined (e.g. for NB probes) then they are logged and reported to the L2 complex Master Core.				
18:8	Reserved.				
7	<b>PrefetcherDis</b> . Read-write. Reset: 0. BIOS: See <a href="#">2.3.3 [Using L2 Cache as General Storage During Boot]</a> . 1=Disable the L2 prefetcher.				
6:4	Reserved.				
3	<b>CacheIcAttrDis</b> . Read-write. Reset: 0. BIOS: See <a href="#">2.3.3 [Using L2 Cache as General Storage During Boot]</a> . 1=Disable inserting IC attributes into the L2.				
2:0	Reserved.				

### 3.22 Core Performance Counter Events

This section provides the performance counter events that may be selected through [MSRC001\\_00\[03:00\]\[EventSelect,UnitMask\]](#). See [MSRC001\\_00\[03:00\]](#) and [MSRC001\\_00\[07:04\]](#) [Performance Event Counter (PERF\_CTR[3:0])].

For L2I performance counter events see [2.6.1.2 \[L2I Performance Monitor Counters\]](#) and [3.23 \[L2I Performance Counter Events\]](#).

For NB performance counter events see [2.6.1.3 \[NB Performance Monitor Counters\]](#) and [3.24 \[NB Performance Counter Events\]](#).

#### 3.22.1 PMCx0[1F:00] Events (FP)

##### **PMCx000 Dispatched FPU Operations**

The number of operations (uops) dispatched to the FPU execution pipelines. This includes all operations done by x87, MMX™ and SSE instructions, including moves. This event is a speculative event. (See PMCx0CB). Since this event includes non-numeric operations it is not suitable for measuring MFLOPs.

UnitMask	Description
7:2	Reserved.
1	Pipe1 dispatches.
0	Pipe0 dispatches.

##### **PMCx001 FP Scheduler Empty**

The number of cycles in which the FPU is empty. Invert this ([MSRC001\\_00\[03:00\]\[Inv\]==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/AVX Operations**

The number of SSE/AVX operations retired. The number of events logged per cycle can vary from 0 to 8.

UnitMask	Description
7	Reserved.
6	Double precision divide/square root FLOPS.
5	Double precision multiply FLOPS.
4	Double precision add/subtract FLOPS.
3	Reserved.
2	Single-precision divide/square root FLOPS.
1	Single-precision multiply FLOPS.



0	Single-precision add/subtract FLOPS.
---	--------------------------------------

### 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. A control-renaming uop requires a rename from a limited pool of control renames.

UnitMask	Description
7:4	Reserved.
3	x87 control-renaming uops retired.
2	x87 bottom-executing uops retired.
1	SSE control-renaming uops retired.
0	SSE bottom-executing uops retired.

### PMCx011 Retired x87 Floating Point Operations

The number of x87 floating point ops that have retired.

UnitMask	Description
7:3	Reserved.
2	Divide and fsqrt ops.
1	Multiply ops.
0	Add/subtract ops.

### 3.22.2 PMCx0[3F:20] Events (LS)

#### 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 that were caused by self-modifying code (a store that hits any instruction that's been fetched for execution beyond the instruction doing the store).

**PMCx022 Pipeline Restart Due to Probe Hit**

The number of pipeline restarts caused by invalidating probes that hit load out-of-order with respect to other load.

UnitMask	Description
7:2	Reserved.
1	Restarts due to evictions caused by fills.
0	Restarts due to evictions caused by invalidating probes.

**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
7:3	Reserved.
2	The number of cycles to unlock \$line (not including cache miss)
1	The number cycles to acquire bus lock
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.

**PMCx029 LS Dispatch**

Counts the number of transactions dispatched to load-store unit.

UnitMask	Description
7:3	Reserved.
2	Load-op-Stores.
1	Stores.
0	Loads.

**PMCx02A Canceled Store to Load Forward Operations**

Counts the number of missed store to load forward opportunities which resulted in blocks.

UnitMask	Description
7:3	Reserved.

2	Block due to store being misaligned.
1	Block due to store being smaller than the load.
0	Block due to address mismatch (starting byte not the same).

### 3.22.3 PMCx0[5F:40] Events (DC)

#### 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 miss in the data cache and allocate a MAB. 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. The UnitMask selects lines in one or more specific coherency states. Each increment reflects a 64-byte transfer. If UnitMask[0] is selected it might be less than 64-bytes. This event is a speculative event.

UnitMask	Description
7:5	Reserved.
4	Modified
3	Owned
2	Exclusive
1	Shared
0	Non-cacheable return of data.

#### 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	Non-cacheable read data.

**PMCx044 Data Cache Lines Evicted**

The UnitMask may be used to count only victims in specific coherency states. Each increment represents a 64-byte transfer. Lines brought into the data cache by PrefetchNTA instructions are evicted directly to system memory (if dirty) or invalidated (if clean). This event is a speculative event.

UnitMask	Description
7:5	Reserved.
4	Modified eviction
3	Owned eviction
2	Exclusive eviction
1	Shared eviction
0	Evicted from probe

**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.

**PMCx046 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
7:4	Reserved.
3	Count loads that miss L2TLB.
2	Count stores that miss L2TLB.
1	Count loads that miss L1TLB.
0	Count stores that miss L1TLB.

**PMCx047 Misaligned Accesses**

The number of data cache accesses that are misaligned. These are accesses which cross the boundary as specified by UnitMask. Misaligned accesses incur at least an extra cache access and an extra cycle of latency on reads. This event is a speculative event.

UnitMask	Description
7:2	Reserved.
1	Number of misaligns that cross a 4kb boundary
0	Number of misaligns that cross a 16-byte boundary

**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 specula-

tive 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 cacheable locked operations that miss in the data cache. Cacheable locks are defined as locks that hit write back memory space and are not misaligned.

### **PMCx04D L1 DTLB Hit**

The number of data cache accesses that hit in the L1 DTLB. This event is a speculative event.

UnitMask	Description
7:2	Reserved.
1	L1 2M TLB hit
0	L1 4K TLB hit

### **PMCx052 Ineffective Software Prefetchs**

The number of software prefetches that do not cause an actual data cache refill. The unit mask may be used to determine the specific cause.

UnitMask	Description
7:4	Reserved.
3	SW Prefetch hit in L2.
2	SW prefetches that don't get a MAB and don't cause <a href="#">PMCx052[1,0]</a> .
1	Software prefetch hit a pending fill.
0	Software prefetch hit in the data cache.

### **PMCx054 Global Page Invalidations**

This event counts TLB flushes that flush TLB entries that have the global bit set.

#### **3.22.4 PMCx[8,1:0][7F:60] Events (BU)**

### **PMCx062 Command Related to Read Block Operations**

UnitMask	Description
7	Reserved.
6	Read Block Speculative Shared.

5	RdBlkSpecMod.
4	RdBlkSpec.
3	Reserved.
2	Read Block Shared.
1	RdBlkMod.
0	Read Block.

### PMCx063 Command Related to Change to Dirty Operations

---

UnitMask	Description
7:5	Reserved.
4	Change to Dirty.
3:0	Reserved.

### PMCx065 Memory Requests by Type

---

These events reflect accesses to memory of each region type (as defined by MTRR or PAT settings). UnitMask[1] and UnitMask[7] reflect full 64 byte writes or full or partial 32 byte writes.

UnitMask	Description
7	Streaming store (SS) requests.
6:2	Reserved.
1	Request to write-combining (WC) memory.
0	Requests to non-cacheable (UC) memory.

### PMCx067 Data Cache Prefetches

---

UnitMask	Description
7:4	Reserved.
3	Prefetch hits on MAB.
2	Reserved.
1	Prefetch attempts.
0	Reserved.

**PMCx068 MAB Requests**

These events reflect utilization of the Miss Address Buffers (MABs), which handle L1 cache misses. The UnitMask is an encoded value which selects one of the IC or DC miss buffers. EventSelect 68h counts the number of misses handled by the selected MAB; EventSelect 69h counts the number of cycles the selected MAB is busy waiting for the refill response.

When used together, these two events provide a measure of the average refill latency seen by the selected MAB, which is the number of cycles spent waiting (EventSelect 69h) divided by the number of requests (EventSelect 68h). This calculation gives the most direct indication of average system latency for Dcache and Icache refills, respectively. For other entries this value typically includes queuing delays caused by resource contention with prior refills.

UnitMask	Description												
7:0	<b>BufferN.</b> where N = 00h to 0Ch. <table border="0"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>07h-00h</td> <td>DC miss buffers 0 to 7</td> </tr> <tr> <td>0Ah-08h</td> <td>IC miss buffers 0 to 2</td> </tr> <tr> <td>0Bh</td> <td>Any DC miss buffer</td> </tr> <tr> <td>0Ch</td> <td>Any IC miss buffer</td> </tr> <tr> <td>FFh-0Dh</td> <td>Reserved</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	07h-00h	DC miss buffers 0 to 7	0Ah-08h	IC miss buffers 0 to 2	0Bh	Any DC miss buffer	0Ch	Any IC miss buffer	FFh-0Dh	Reserved
<u>Bits</u>	<u>Definition</u>												
07h-00h	DC miss buffers 0 to 7												
0Ah-08h	IC miss buffers 0 to 2												
0Bh	Any DC miss buffer												
0Ch	Any IC miss buffer												
FFh-0Dh	Reserved												

**PMCx069 MAB Wait Cycles**

See [PMCx068](#).

UnitMask	Description
7:0	<b>BufferID.</b> See: <a href="#">PMCx068</a> [BufferN].

**PMCx06C System Response by Coherence State**

The number of responses from the L2I for cache refill requests. The UnitMask may be used to select specific cache coherency states. Each of {Exclusive, Modified, Shared} represents one 64-byte cache line transferred from the L2I (DRAM, L2 cache, or the L1 cache of another core) to the data cache or instruction cache.

- Modified responses may be for Dcache store miss refills, PrefetchW software prefetches, or Change-to-Dirty requests that get a dirty probe hit in another cache.
- Exclusive responses may be for:
  - Dcache load miss refill.
  - Software prefetches.
- Shared responses may be for:
  - Icache refill.
  - Any request that hits a clean line in another cache.
- Change-to-Dirty success response is for a Dcache upgrade request (store hit to a shared line).
- Uncacheable response is for all uncacheable system requests.

UnitMask	Description
7	Reserved.

6	Uncacheable.
5	Change-to-Dirty success.
4	Data Error.
3	Owned.
2	Shared.
1	Modified.
0	Exclusive.

### PMCx06D Data Written to System

The number of 16-byte data transfers from the processor to the system. Note that a dirty cache line writeback would cause four increments. An IO write would cause two increments: one for the mask and one for the data.

UnitMask	Description
7:6	Reserved.
5	Data for Write Combine context flush.
4	Byte Enable Mask for Write Combine context flush.
3	Data for uncacheable or IO store.
2	Byte Enable Mask for uncacheable or IO store.
1	Instruction attribute evictions.
0	Data line evictions.

### PMCx075 Cache Cross-invalidates

These reflect internal probes for Icache or Dcache misses that hit in the Dcache or Icache, causing the line to be invalidated. These may result from code modification, data being located too close to code, or virtual address aliasing. The aliasing cases arise when a physical memory location is referenced via two or more virtual addresses which differ in bits 14:12. Such aliasing cases are generally uncommon.

UnitMask	Description
7	Reserved.
6	DC probe rejected late.
5	DC probe rejected early.
4	IC hits DC Clean Line (reading code).
3	IC Invalidates DC Dirty Line (execution of recently modified code, or modified data too close to code).
2	IC Invalidates IC (aliasing).
1	DC Invalidates DC (CD or WBINVD).
0	DC Invalidates IC (modification of cached instructions, or of data located too close to code).

### PMCx076 CPU Clocks not Halted

The number of clocks that the CPU is not in a halted state (due to STPCLK or a HLT instruction). Note: 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: Reserved. See [L2IPMCx07D \[Requests to L2 Cache\]](#).

PMCx07E: Reserved. See [L2IPMCx07E \[L2 Cache Misses\]](#).

PMCx07F: Reserved. See [L2IPMCx07F \[L2 Fill/Writeback\]](#).

### PMCx162 PDC Miss

Counts the number of PDC miss events specified by the UnitMask.

UnitMask	Description
7	Reserved.
6	Guest: PML4E Level.
5	Guest: PDPE Level.
4	Guest: PDE Level.
3	Reserved.
2	Host: PML4E Level.
1	Host: PDPE Level.
0	Host: PDE Level.

### 3.22.5 PMCx[1:0][9F:80] Events (IC)

Note: All instruction cache events are speculative events unless specified otherwise.

#### PMCx080 Instruction Cache Fetches

The number of successful instruction cache accesses by the instruction fetcher that result in data being sent to the decoder. 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 ITLB Miss**

The number of instruction fetches that miss in the 4K ITLB and 2M ITLB.

UnitMask	Description
7:2	Reserved.
1	Instruction fetches to a 2M page.
0	Instruction fetches to a 4K page.

**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
7:2	Reserved.
1	IC invalidate due to a BU probe.

0	IC invalidate due to an LS probe.
---	-----------------------------------

### PMCx099 ITLB Reloads

The number of ITLB reload requests.

### PMCx09A ITLB Reloads Aborted

The number of ITLB reloads aborted.

### PMCx19A Retired Indirect Branch Info

UnitMask	Description
7:2	Reserved.
1	Retired Mispredicted Indirect Branch Instruction. Retired Mispredicted Near Unconditional jump with target specified in Register or Memory. This event is non-speculative.
0	Retired Indirect branch Instruction. Retired Near Unconditional jump with target specified in Register or Memory. This event is non-speculative.

### 3.22.6 PMCx[1,0][DF:C0] Events (EX, DE)

#### 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 were not correctly predicted in either target or direction. This includes those for which prediction is not attempted (far control transfers, exceptions and interrupts), and excludes resyncs.

#### PMCx0C4 Retired Taken Branch Instructions

The number of taken branches that were retired. This includes all types of architectural control flow changes, including exceptions and interrupts, and excludes resyncs.

**PMCx0C5 Retired Taken Branch Instructions Mispredicted**

The number of retired taken branch instructions that were mispredicted, and excludes resyncs.

**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, and excludes resyncs. 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**

A near return instruction was retired that mispredicted in either target or direction.

**PMCx0CA Retired Mispredicted Taken Branch Instructions due to Target Mismatch**

A taken branch instruction was retired that mispredicted in target address (but not in direction).

**PMCx0CB Retired MMX™/FP Instructions**

A floating point (x87, MMX™, or SSE) instruction was retired. The UnitMask allows the distinction between x87/MMX™ and SSE instructions. Since this event includes non-numeric instructions it is not suitable for measuring MFLOPS.

UnitMask	Description
7:2	Reserved.
1	SSE floating point instruction was retired (SSE, SSE2, SSE3, MNI)
0	x87 or MMX™ instruction was retired

**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).

### **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

### **PMCx0D[F:C] DR[3:0] Breakpoint Matches**

Table 230: [Register Mapping](#) for **PMCx0D[F:C]**

Register	Function
PMCx0DC	DR0
PMCx0DD	DR1
PMCx0DE	DR2
PMCx0DF	DR3

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.

### **PMCx1CF Tagged IBS Ops**

Number of ops tagged by IBS.

### **PMCx1D0 Tagged IBS Ops Retired**

The number of retired tagged IBS operations.

### 3.23 L2I Performance Counter Events

#### L2IPMCx060 Command Related to Victim Buffers

UnitMask	Description
7	<b>Lock.</b>
6:5	Reserved.
4	<b>ClVicBlk: clean victim command.</b>
3	<b>WrVicBlkNR: write victim block non-rinsing.</b>
2	<b>WrVicBlkR: write victim block rinsing.</b>
1:0	Reserved.

#### L2IPMCx061 Command Related to Masked Operations

Count Masked Byte and DW reads and writes to the NB. Byte sized read and write commands can request the transfer of up to 32 bytes. DW sized read and write commands can request the transfer of up to 32 DW's. Combining of the WC memory type can cause 1 B/DW write to represent multiple stores.

UnitMask	Description
7:6	Reserved.
5	<b>WrDWord: write double-word.</b>
4	<b>WrByte: write byte.</b>
3	Reserved.
2	<b>RdDWord: read double-word.</b>
1	Reserved.
0	<b>RdByte: read-byte.</b>

#### L2IPMCx062 Command Related to Read Block Operations

UnitMask	Description
7:3	Reserved.
2	<b>RdBlkS: read block shared.</b>
1	<b>RdBlkM: read block modified.</b>
0	<b>RdBlk: read block.</b>

#### L2IPMCx063 Command Related to Change to Dirty Operations

UnitMask	Description
7:5	Reserved.
4	<b>ChgToDirty: change to dirty.</b>
3:0	Reserved.

**L2IPMCx06C Response From System on Cache Refills**

The number of responses from the system for cache refill requests. The UnitMask may be used to select specific cache coherency states. Each increment represents one 64 B cache line transferred from the system (DRAM or another cache, including another core on the same node) 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	Reserved.
6	<b>Uncacheable.</b>
5	<b>ChgToDirtySuccess.</b>
4	<b>DataError.</b>
3	<b>Owned.</b>
2	<b>Shared.</b>
1	<b>Modified.</b>
0	<b>Exclusive.</b>

**L2IPMCx07D Requests to L2 Cache**

UnitMask	Description
7	Reserved.
6	<b>PrefReq.</b> L2 cache prefetcher request.
5:4	Reserved.
3	<b>NbPrbReq.</b> NB probe request.
2	Reserved.
1	<b>DcFill.</b> DC fill.
0	<b>IcFill.</b> IC fill.

**L2IPMCx07E L2 Cache Misses**

The number of requests that miss in the L2 cache. This may include some amount of speculative activity. The IC-fill-miss and DC-fill-miss events tend to mirror the Icache and Dcache refill-from-system [PMCx083](#) and [PMCx043](#), and tend to include more speculative activity than those events.

UnitMask	Description
7:5	Reserved.
4	<b>L2PrefReq.</b> L2 Cache Prefetcher request
3:2	Reserved.
1	<b>DcFill.</b>
0	<b>IcFill.</b>

**L2IPMCx07F L2 Fill/Writeback**

UnitMask	Description
7:3	Reserved.
2	<b>L2CleanWritebacks.</b> L2 Clean Writebacks to system.
1	<b>L2Writebacks.</b> L2 Writebacks to system (Clean and Dirty).
0	<b>L2Fills.</b> L2 fills from system. Note: Fills for non-temporal software prefetch and WP-memtype fills also are counted in this event even though they don't get cached in L2.

**3.24 NB Performance Counter Events**

This section provides the performance counter events that may be selected through [MSRC001\\_024\[6,4,2,0\]\[EventSelect,UnitMask\]](#). See that register and [MSRC001\\_024\[7,5,3,1\]](#) [Northbridge Performance Event Counter (NB\_PERF\_CTR[3:0])].

**3.24.1 PMCx0E[7:4] Events (Memory Controller)****NBPMCx0E4 Memory Controller Bypass Counter Saturation**

UnitMask	Description																		
7:6	Reserved.																		
5	DCQ Bypass Saturated. The DCT is selected by the field NBPMCx0E4[4:2].																		
4:2	Select DCQ bypass: <table border="1" data-bbox="305 1108 876 1423"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>Select DCT0 DCQ bypass</td> </tr> <tr> <td>001b</td> <td>Select DCT1 DCQ bypass</td> </tr> <tr> <td>010b</td> <td>Select DCT2 DCQ bypass</td> </tr> <tr> <td>011b</td> <td>Select DCT3 DCQ bypass</td> </tr> <tr> <td>100b</td> <td>Select DCT4 DCQ bypass</td> </tr> <tr> <td>101b</td> <td>Select DCT5 DCQ bypass</td> </tr> <tr> <td>110b</td> <td>Select DCT6 DCQ bypass</td> </tr> <tr> <td>111b</td> <td>Select DCT7 DCQ bypass</td> </tr> </tbody> </table>	Bits	Description	000b	Select DCT0 DCQ bypass	001b	Select DCT1 DCQ bypass	010b	Select DCT2 DCQ bypass	011b	Select DCT3 DCQ bypass	100b	Select DCT4 DCQ bypass	101b	Select DCT5 DCQ bypass	110b	Select DCT6 DCQ bypass	111b	Select DCT7 DCQ bypass
Bits	Description																		
000b	Select DCT0 DCQ bypass																		
001b	Select DCT1 DCQ bypass																		
010b	Select DCT2 DCQ bypass																		
011b	Select DCT3 DCQ bypass																		
100b	Select DCT4 DCQ bypass																		
101b	Select DCT5 DCQ bypass																		
110b	Select DCT6 DCQ bypass																		
111b	Select DCT7 DCQ bypass																		
1	Memory controller medium priority bypass																		
0	Memory controller high priority bypass																		

**3.24.2 PMCx0E[F:8] Events (Crossbar)****NBPMCx0E8 Thermal Status**

UnitMask	Description
7	Reserved
6	Number of clocks HTC P-state is active
5	Number of clocks HTC P-state is inactive



4	Reserved.
3	Reserved.
2	Number of times the HTC trip point is crossed
1:0	Reserved

### **NBPMCx0E9 CPU/IO Requests to Memory/IO**

These events reflect request flow between units and nodes, as selected by the UnitMask. The UnitMask is divided into two fields: request type (CPU or IO access to IO or Memory) and source/target location (local vs. remote). One or more requests types must be enabled via bits 3:0, and at least one source and one target location must be selected via bits 7:4. Each event reflects a request of the selected type(s) going from the selected source(s) to the selected target(s).

Not all possible paths are supported. The following table shows the UnitMask values that are valid for each request type:

Source/Target	CPU to Mem	CPU to IO	IO to Mem	IO to IO
Local -> Local	A8h	A4h	A2h	A1h
Local -> Remote	98h	94h	92h	91h
Remote -> Local	-	64h	-	61h
Remote -> Remote	-	-	-	-

Any of the mask values shown may be logically ORed to combine the events. For instance, local CPU requests to both local and remote nodes would be  $A8h | 98h = B8h$ . Any CPU to any IO would be  $A4h | 94h | 64h = F4h$  (but remote CPU to remote IO requests would not be included).

Note: 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. For a direct measure of the amount and direction of data flowing between nodes, use events F6h, F7h and F8h.

UnitMask	Description
7	From local node
6	From remote node
5	To local node
4	To remote node
3	CPU to Mem
2	CPU to IO
1	IO to Mem
0	IO to IO

### **NBPMCx0EA 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)

### NBPMCx0EB 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. Typical uses of the various Sized Read/Write commands are given in the UnitMask table. See [NBPMCx0EC](#), which provides a separate measure of Hostbridge accesses.

UnitMask	Description
7:6	Reserved.
5	SzRd DW (1-16 DWORDs). Typical Usage: Block-oriented DMA reads, typically cache-line size.
4	SzRd Byte (4 bytes). Typical Usage: Legacy or mapped IO.
3	Posted SzWr DW (1-16 DWORDs). Typical Usage: Block-oriented DMA writes, often cache-line sized; also processor Write Combining buffer flushes.
2	Posted SzWr Byte (1-32 bytes). Typical Usage: Sub-cache-line DMA writes, size varies; also flushes of partially-filled Write Combining buffer.
1	Non-Posted SzWr DW (1-16 DWORDs). Typical Usage: Legacy or mapped IO, typically 1 DWORD.
0	Non-Posted SzWr Byte (1-32 bytes). Typical Usage: Legacy or mapped IO, typically 1-4 bytes.

### NBPMCx0EC Probe Responses and Upstream Requests

This covers two unrelated sets of events: cache probe results, and requests received by the hostbridge from devices on non-coherent links.

**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 requestor (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 IO link.

UnitMask	Description
7	Upstream non-ISOC writes
6	Upstream ISOC writes
5	Upstream non-display refresh reads
4	Upstream display refresh/ISOC 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

### 3.24.3 PMCx0F[F:0] Events (Crossbar)

### 3.24.4 NBPMCx1E[F:0] Events (Crossbar)

#### **NBPMCx1E0 CPU to DRAM Requests to Target Node**

This event counts all DRAM reads and writes generated by cores on the local node to the targeted node in the coherent fabric. This counter can be used to observe processor data affinity in NUMA aware operating systems.

UnitMask	Description
7	From Local node to Node 7
6	From Local node to Node 6
5	From Local node to Node 5
4	From Local node to Node 4
3	From Local node to Node 3
2	From Local node to Node 2
1	From Local node to Node 1
0	From Local node to Node 0

#### **NBPMCx1E1 IO to DRAM Requests to Target Node**

This event counts all DRAM reads and writes generated by IO devices attached to the IO links of the local node the targeted node in the coherent fabric. This counter can be used to observe IO device data affinity in NUMA aware operating systems.

UnitMask	Description
7	From Local node to Node 7
6	From Local node to Node 6
5	From Local node to Node 5
4	From Local node to Node 4
3	From Local node to Node 3
2	From Local node to Node 2

1	From Local node to Node 1
0	From Local node to Node 0

### **NBPMCx1E2 CPU Read Command Latency to Target Node 0-3**

This event counts the number of NB clocks from when the targeted command is received in the NB to when the targeted command completes. This event only tracks one outstanding command at a time. To determine latency between the local node and a remote node set UnitMask[7:4] to select the node and UnitMask[3:0] to select the read command type. The count returned by the counter should be divided by the count returned by [NBPMCx1E3](#) do determine the average latency for the command type.

UnitMask	Description
7	From Local node to Node 3
6	From Local node to Node 2
5	From Local node to Node 1
4	From Local node to Node 0
3	Change-to-Dirty
2	Read block modified
1	Read block shared
0	Read block

### **NBPMCx1E3 CPU Read Command Requests to Target Node 0-3**

This event counts the number of requests that a latency measurement is made for using [NBPMCx1E2](#). To determine the number of commands that a latency measurement are made for between the local node and a remote node set UnitMask[7:4] to select the node and UnitMask[3:0] to select the read command type.

UnitMask	Description
7	From Local node to Node 3
6	From Local node to Node 2
5	From Local node to Node 1
4	From Local node to Node 0
3	Change-to-Dirty
2	Read block modified
1	Read block shared
0	Read block

### **NBPMCx1E4 CPU Read Command Latency to Target Node 4-7**

This event counts the number of NB clocks from when the targeted command is received in the NB to when the targeted command completes. This event only tracks one outstanding command at a time. To determine latency between the local node and a remote node set UnitMask[7:4] to select the node and UnitMask[3:0] to select the read command type. The count returned by the counter should be divided by the count returned by [NBPMCx1E5](#) do determine the average latency for the command type.

UnitMask	Description
7	From Local node to Node 7
6	From Local node to Node 6
5	From Local node to Node 5
4	From Local node to Node 4
3	Change-to-Dirty
2	Read block modified
1	Read block shared
0	Read block

### **NBPMCx1E5 CPU Read Command Requests to Target Node 4-7**

This event counts the number of requests that a latency measurement is made for using [NBPMCx1E4](#). To determine the number of commands that a latency measurement are made for between the local node and a remote node set UnitMask[7:4] to select the node and UnitMask[3:0] to select the read command type.

UnitMask	Description
7	From Local node to Node 7
6	From Local node to Node 6
5	From Local node to Node 5
4	From Local node to Node 4
3	Change-to-Dirty
2	Read block modified
1	Read block shared
0	Read block

### **NBPMCx1E6 CPU Command Latency to Target Node 0-3/4-7**

This event counts the number of NB clocks from when the targeted command is received in the NB to when the targeted command completes. This event only tracks one outstanding command at a time. To determine latency between the local node and a remote node set UnitMask[7:4] to select the node, UnitMask[3] to select the node group and UnitMask[3:0] to select the command type. The count returned by the counter should be divided by the count returned by [NBPMCx1E7](#) do determine the average latency for the command type.

UnitMask	Description
7	From Local node to Node 3/7
6	From Local node to Node 2/6
5	From Local node to Node 1/5
4	From Local node to Node 0/4
3	Node Group Select. 0=Nodes 0-3. 1= Nodes 4-7.
2	Victim Block
1	Write Sized
0	Read Sized

**NBPMCx1E7 CPU Requests to Target Node 0-3/4-7**

This event counts the number of requests that a latency measurement is made for using [NBPMCx1E6](#). To determine the number of commands that a latency measurement are made for between the local node and a remote node set UnitMask[7:4] to select the node, UnitMask[3] to select the node group and UnitMask[3:0] to select the command type.

UnitMask	Description
7	From Local node to Node 3/7
6	From Local node to Node 2/6
5	From Local node to Node 1/5
4	From Local node to Node 0/4
3	Node Group Select. 0=Nodes 0-3. 1= Nodes 4-7.
2	Victim Block
1	Write Sized
0	Read Sized

**NBPMCx1EB Request Cache Status 1**

The probe response type for RdBlkM or ChgToDirty request type.

UnitMask	Description
7	Track Cache Stat for RdBlkM
6	Track Cache Stat for ChgToDirty
5	Directed Probe
4	Probe Miss
3	Probe Hit M
2	Probe Hit MuW or O
1	Probe Hit E
0	Probe Hit S

**3.24.5 NBPMCx1F[F:0] Events (Memory Controller, Crossbar)****NBPMCx1F0 Memory Controller Requests**

Read/Write requests: The read/write request events reflect the total number of commands sent to the DRAM controller.

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 (on any node in an MP system). 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. [PMCx065](#) 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) gives an indication of how efficiently the write combining buffers are being used. [PMCx065](#) may also be useful in factoring out WC flushes when comparing these events with the Upstream Requests component of [PMCx06C](#).

UnitMask	Description
7	Read requests sent to the DCT while writes requests are pending in the DCT
6	64 Byte Sized Reads
5	32 Bytes Sized Reads
4	64 Bytes Sized Writes
3	32 Bytes Sized Writes
2	Prefetch requests sent to the DCT
1	Read requests (including prefetch requests) sent to the DCT
0	Write requests sent to the DCT

### **NBPMCx3EC DRAM Accesses**

The number of memory accesses performed by the local DRAM controller. UnitMask[7:0] 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)

Each DRAM access represents one 64-byte block of data transferred if the DRAM is configured for 64-byte granularity, or one 32-byte block if the DRAM is configured for 32-byte granularity. (The latter is only applicable to single-channel DRAM systems, which may be configured either way.)

UnitMask	Description
7:6	Reserved.
5	DCT1 Page Conflict
4	DCT1 Page Miss
3	DCT1 Page hit
2	DCT0 Page Conflict
1	DCT0 Page Miss
0	DCT0 Page hit

### **NBPMCx3ED DRAM Controller Page Table Overflows**

The number of page table overflows 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 maximum number of pages are already open. Each occurrence reflects an access latency penalty equivalent to a page conflict.

UnitMask	Description
7:2	Reserved.

1	DCT1 Page Table Overflow
0	DCT0 Page Table Overflow

**NBPMCx3EE Memory Controller DRAM Command Slots Missed**

UnitMask	Description
7:2	Reserved.
1	DCT1 Command Slots Missed (in MEMCLKs)
0	DCT0 Command Slots Missed (in MEMCLKs)

**NBPMCx3EF Memory Controller Turnarounds**

The number of turnarounds on the local DRAM data bus. UnitMask[7:0] 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:  $DRAM\_width\_in\_bytes * 2 \text{ edges\_per\_memclk} * 2$
- R/W turnaround:  $DRAM\_width\_in\_bytes * 2 \text{ edges\_per\_memclk} * 1$
- R/W turnaround:  $DRAM\_width\_in\_bytes * 2 \text{ edges\_per\_memclk} * (Tcl-1)$

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
7:6	Reserved.
5	DCT1 Write to read turnaround
4	DCT1 Read to write turnaround
3	DCT1 DIMM (chip select) turnaround
2	DCT0 Write to read turnaround
1	DCT0 Read to write turnaround
0	DCT0 DIMM (chip select) turnaround



### 3.25 Fusion Controller Hub Registers

#### 3.25.1 Legacy Block Configuration Registers (IO)

See 3.1 [Register Descriptions and Mnemonics] for a description of the register naming convention.

##### IO000 Dma\_Ch 0

Bits	Description
15:0	<b>DmaCh0</b> . Read-write. Reset: 0. DMA1 channel 0 base and current address.

##### IO002 Dma\_Ch 1

Bits	Description
15:0	<b>DmaCh1</b> . Read-write. Reset: 0. DMA1 channel 1 base and current address.

##### IO004 Dma\_Ch 2

Bits	Description
15:0	<b>DmaCh2</b> . Read-write. Reset: 0. DMA1 channel 2 base and current address.

##### IO006 Dma\_Ch 3

Bits	Description
15:0	<b>DmaCh3</b> . Read-write. Reset: 0. DMA1 channel 3 base and current address.

##### IO008 Dma\_Status

Bits	Description
7:0	<b>DmaStatus</b> . Read-write. Reset: 0. Specifies the DMA status register for channels 0-3 for reads; specifies the DMA control register for channels 0-3 for writes.

##### IO009 Dma\_WriteRequest

Bits	Description
7:0	<b>DmaWriteRequest</b> . Read-write. Reset: 0. Request register.

##### IO00A Dma\_WriteMask

Bits	Description
7:0	<b>DmaWriteMask</b> . Read-write. Reset: 0. DMA channel mask register.

**IO00B Dma\_WriteMode**

Bits	Description
7:0	<b>DmaWriteMode</b> . Read-write. Reset: 0. DMA mode register.

**IO00C Dma\_Clear**

Bits	Description
7:0	<b>DmaClear</b> . Read-write. Reset: 0. Channel 0-3 DMA clear byte pointer.

**IO00D Dma\_MasterClr**

Dma\_MasterClr register

Bits	Description
7:0	<b>DmaMasterClr</b> . Read-write. Reset: 0. Write: Channel 0-3 master clear register. Read: Intermediate register.

**IO00E Dma\_ClrMask**

Bits	Description
7:0	<b>DmaClrmask</b> . Read-write. Reset: 0. Channel 0-3 DMA clear mask.

**IO00F Dma\_AllMask**

Bits	Description
7:0	<b>DmaAllMask</b> . Read-write. Reset: 0. General mask register.

**IO020 IntrCntrl1Reg1**

Bits	Description
7:0	<b>IntrCntrl1Reg1</b> . Read-write. Reset: 0. IRQ0-7 status and control. Read: IRR, ISR. Write: ICW1, OCW2, OCW3.

**IO021 IntrCntrl1Reg2**

Bits	Description
7:0	<b>IntrCntrl1Reg2</b> . Read-write. Reset: 0. IRQ0-7 status and control. Read: IMR. Write: ICW2, ICW3, ICW4, OCW1.

**IO022 IMCR\_Index**

Bits	Description
7:0	<b>ImcrIndex</b> . Read-write. Reset: 0. Index port for IMCR register. See <a href="#">IO023 [IMCR_Data]</a> .

**IO023 IMCR\_Data**

Bits	Description
7:0	<b>ImcrData</b> . Read-write. Reset: 0. Data port for IMCR register. The actual IMCR register is located at index 70h and it is at bit 0.

**IO040 TimerCh0**

Bits	Description
7:0	<b>TimerCh0</b> . Read-write. Reset: 0. 8254 Timer 1 Counter 0 Data Port.

**IO041 TimerCh1**

Bits	Description
7:0	<b>TimerCh1</b> . Read-write. Reset: 0. 8254 Timer 1 Counter 1 Data Port.

**IO042 TimerCh2**

Bits	Description
7:0	<b>TimerCh2</b> . Read-write. Reset: 0. 8254 Timer 1 Counter 2 Data Port.

**IO043 Tmr1CntlWord**

Bits	Description										
7:6	<b>CounterSelect</b> . Read-write. Reset: 0. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Select counter 0</td> </tr> <tr> <td>01b</td> <td>Select counter 1</td> </tr> <tr> <td>10b</td> <td>Select counter 2</td> </tr> <tr> <td>11b</td> <td>Read back command</td> </tr> </tbody> </table>	Bits	Definition	00b	Select counter 0	01b	Select counter 1	10b	Select counter 2	11b	Read back command
Bits	Definition										
00b	Select counter 0										
01b	Select counter 1										
10b	Select counter 2										
11b	Read back command										
5:4	<b>CommandSelect</b> . Read-write. Reset: 0. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Counter latch command</td> </tr> <tr> <td>01b</td> <td>Read/write least significant byte</td> </tr> <tr> <td>10b</td> <td>Read/write most significant byte</td> </tr> <tr> <td>11b</td> <td>Read/write least, and then most significant byte</td> </tr> </tbody> </table>	Bits	Definition	00b	Counter latch command	01b	Read/write least significant byte	10b	Read/write most significant byte	11b	Read/write least, and then most significant byte
Bits	Definition										
00b	Counter latch command										
01b	Read/write least significant byte										
10b	Read/write most significant byte										
11b	Read/write least, and then most significant byte										

3:1	<b>ModeSelect.</b> Read-write. Reset: 0. <u>Bits</u> 000b 001b 010b 011b 100b 101b 111b-110b	<u>Definition</u> Asserts OUT signal at end of count Hardware re-triggerable one-shot Rate generator Square wave output Software triggered strobe Hardware triggered strobe Reserved
0	<b>CntDownSelect.</b> Read-write. Reset: 0. 0=Binary countdown. 1=BCD countdown.	

**IO060 Keyboard Data**

Bits	Description
7:0	<b>Data.</b> Read-write. Reset: 0. Specifies the keyboard controller data port. It should only be read from when <a href="#">IO064[0]</a> reads back 1. It should only be written to when <a href="#">IO064[1]</a> reads back 0.

**IO061 Nmi Status**

Bits	Description
7	<b>ParityErrNmi.</b> Read-only. Reset: X. NMI is caused by parity error (either PERR# or SERR#).
6	<b>IoChkNmi.</b> Read-only. Reset: X. NMI is triggered by serial IOCHK.
5	<b>SpkrClk.</b> Read-only. Reset: X. The output of 8254 timer counter 2.
4	<b>RefClk.</b> Read-only. Reset: X. The output of 8254 timer counter 1.
3	<b>IoChkNmiEn.</b> Read-write. Reset: 1. 0=Enable IoChk to NMI generation. 1=Disable IoChk to NMI generation.
2	<b>ParityErrNmiEn.</b> Read-write. Reset: 1. 0=Enable Parity Error to NMI generation (from SERR# or PERR#). 1=Disable Parity Error to NMI generation and clear bit 7.
1	<b>SpkrTmrEnable.</b> Read-write. Reset: 0. 0=Speaker timer off. 1=Speaker timer on.
0	<b>SpkrEnable.</b> Read-write. Reset: 0. 0=Disable counter 2. 1=Enable counter 2.

**IO064 Keyboard Status/Control**

IF (READ)

Bits	Description
7	<b>KeyPar.</b> Read-only. Reset: 0. 1=Keyboard data transmission parity error.
6	<b>KeyRTO.</b> Read-only. Reset: 0. 1=Receive time out.
5	<b>KeyTTO.</b> Read-only. Reset: 0. 1=Keyboard transmit time out or mouse data available.
4	<b>KeyEnB.</b> Read-only. Reset: 0. 0=Password protected.
3	<b>CmdData.</b> Read-only. Reset: 0. 1=Between the keyboard command and data ports, the command port at 64h was last written to. 0=Between the keyboard command and data ports, the data port at 60h was last written to.
2	<b>KeySys.</b> Read-only. Reset: 0. Specifies system flag status. 1=Power up or reset. 1=Soft reset.

1	<b>KeyIBF</b> . Read-only. Reset: 0. 1=Input buffer full. 0=Ready to receive next command/data.
0	<b>KeyOBF</b> . Read-only. Reset: 0. 1=Data is available to read back. 0=No data available.

ELSE

Bits	Description
7:0	<b>KeyCmd</b> . Write-only. Reset: 0. Specified the keyboard command.

ENDIF.

**IO070 RtcAddrPort and NmiMask**

Bits	Description
7	<b>NmiMask</b> . Write-only. Reset: 0. 0=NMI enabled. 1=NMI masked.
6:0	<b>RtcAddrPort: RTC Address Port</b> . Read-write. Reset: 0. This is used with either internal RTC or external RTC. This port specifies the index to access RTC time registers and the CMOS RAM space. <a href="#">IO073_x0A[DV0]</a> should be programmed first to select bank when accessing the CMOS RAM space. See <a href="#">IO073_x00</a> .

**IO071 RtcDataPort**

Bits	Description
7:0	<b>RtcDataPort: RTC Data Port</b> . Read-write. Reset: 0. This is used with either internal RTC or external RTC in conjunction with <a href="#">IO070</a> .

**IO072 Alternate RTC AddrPort**

Bits	Description
7:0	<b>AlternatRTCAddrPort</b> . Read-write. Reset: 0. This is used with internal RTC. This port allows user to specify the full 8 bit address (instead of bank0/bank1 indexing) to access the 256 bytes of RTC RAM.

**IO073 Alternate RTC DataPort**

Bits	Description
7:0	<b>AlternatRTCDataPort</b> . Read-write. Reset: 0. This is used with internal RTC in conjunction with <a href="#">IO072</a> .

**IO073\_x00 RTC Seconds**

Bits	Description
7:0	<b>Seconds</b> . Read-write. Reset: 0. <b>BCD</b> format. The value range for this register is 00 through 59. If ( <a href="#">IO073_x0B[SET]</a> =1), this register is programmed by software and hardware updating is disabled. If ( <a href="#">IO073_x0B[SET]</a> =0), this register is updated by the RTC logic once per second.

**IO073\_x01 RTC Seconds Alarm**

Bits	Description
7:0	<b>SecondsAlarm: Seconds Alarm.</b> Read-write. Reset: 0. <b>BCD</b> format. If (IO073_x0B[SET]=1), the Seconds Alarm register will never match with IO073_x00 [RTC Seconds]. Else, if (bits [7:6] = 11b) the RTC Seconds Alarm register always matches with RTC Seconds register and causes an RTC alarm event to be generated once per second. See IO073_x0C[AF].

**IO073\_x02 RTC Minutes**

Bits	Description
7:0	<b>Minutes.</b> Read-write. Reset: 0. <b>BCD</b> format. The value range for this register is 00 through 59. If (IO073_x0B[SET]=1), this register programmed by software and hardware updating is disabled. If (IO073_x0B[SET]=0), this register is updated by the RTC logic once per second.

**IO073\_x03 RTC Minutes Alarm**

Bits	Description
7:0	<b>MinutesAlarm: Minutes Alarm.</b> Read-write. Reset: 0. <b>BCD</b> format. If (IO073_x0B[SET]=1), the RTC Minutes Alarm register will never match with IO073_x02 [RTC Minutes]. Else, if (bits [7:6]=11b), the RTC Minutes Alarm register always matches with RTC Minutes register and causes an RTC alarm event to be generated once per minute. See IO073_x0C[AF].

**IO073\_x04 RTC Hours**

Bits	Description
7:0	<b>Hours.</b> Read-write. Reset: 0. <b>BCD</b> format. The value range for this register is 00 through 23. If (IO073_x0B[SET]=1), this register can be set by software and hardware updating is disabled. If (IO073_x0B[SET]=0), this register is updated by the RTC logic once per second.

**IO073\_x05 RTC Hours Alarm**

Bits	Description
7:0	<b>HoursAlarm: Hours Alarm.</b> Read-write. Reset: 0. <b>BCD</b> format. If (IO073_x0B[SET]=1), the RTC Hours Alarm register will never match with IO073_x04 [RTC Hours]. Else, if (bits [7:6] = 11b) the RTC Hours Alarm register always matches with the RTC Hours register and causes an RTC alarm event to be generated once per hour. See IO073_x0C[AF].

**IO073\_x06 RTC Day of Week**

Bits	Description
7:0	<b>DayOfWeek: Day of Week.</b> Read-write. Reset: 0. <b>BCD</b> format. The value range for this register is 01 through 07 (Sunday = 1). Leap year correction is performed by software. If (IO073_x0B[SET]==1), this register is programmed by software and hardware updating is disabled. If (IO073_x0B[SET]==0), this register is updated by hardware.

**IO073\_x07 RTC Date of Month**

Bits	Description
7:0	<b>DateOfMonth: Date of Month.</b> Read-write. Reset: 0. <b>BCD</b> format. The range is 01 through 31. Leap year correction is performed by software. If (IO073_x0B[SET]==1), this register is programmed by software and hardware updating is disabled. If (IO073_x0B[SET]==0), this register is updated by hardware.

**IO073\_x08 RTC Month**

Bits	Description
7:0	<b>Month.</b> Read-write. Reset: 0. <b>BCD</b> format. The range is 01 through 12. Leap year correction is performed by software. If (IO073_x0B[SET]=1), this register is programmed by software and hardware updating is disabled. If (IO073_x0B[SET]=0), this register is updated by hardware.

**IO073\_x09 RTC Year**

Bits	Description
7:0	<b>Year.</b> Read-write. Reset: 0. <b>BCD</b> format. Range is 00 through 99. No leap year correction capability. Leap year correction has to be done by software. If (IO073_x0B[SET]=1), this register is programmed by software and hardware updating is disabled. If (IO073_x0B[SET]=0), this register is updated by hardware.

**IO073\_x0A RTC Register A**

Bits	Description
7	<b>UIP: Update In Progress.</b> Read-only. Reset: 0. If (IO073_x0B[SET]=1), UIP is cleared. 1=The update transfer will soon occur. 0=The update transfer will not occur for at least 244 us.
6:5	Reserved.

4	<p><b>DV0: Bank Selection.</b> Read-write. Reset: 0. 0=Select bank 0 when accessing the RTC CMOS RAM space through <a href="#">IO070</a> and <a href="#">IO071</a>. 1=Select bank 1 when accessing the RTC CMOS RAM space through <a href="#">IO070</a> and <a href="#">IO071</a>. The FCH has an alternate way to access the RAM without the use of bank select bit. <a href="#">IO072 [Alternate RTC AddrPort]</a> and <a href="#">IO073 [Alternate RTC DataPort]</a> provides indexed access to the full 256 bytes of RAM.</p>																																		
3:0	<p><b>RS[3:0]: Rate Selection.</b> Read-write. Reset: 0. These four rate selection bits select one of the 13 taps on the 15-stage frequency divider or disable the divider output (flat output signal). The tap selected can be used to generate a periodic interrupt. See the following table for the frequency selection.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Tap Frequency (Interrupt Rate)</th> </tr> </thead> <tbody> <tr><td>0000b</td><td>Flat Signal (None)</td></tr> <tr><td>0001b</td><td>256 Hz (3.90625 ms)</td></tr> <tr><td>0010b</td><td>128 Hz (7.8125 ms)</td></tr> <tr><td>0011b</td><td>8.192 kHz (122.070 us)</td></tr> <tr><td>0100b</td><td>4.096 kHz (244.141 us)</td></tr> <tr><td>0101b</td><td>2.048 kHz (488.281 us)</td></tr> <tr><td>0110b</td><td>1.024 kHz (976.5625 us)</td></tr> <tr><td>0111b</td><td>512 Hz (1.953125 ms)</td></tr> <tr><td>1000b</td><td>256 Hz (3.90625 ms)</td></tr> <tr><td>1001b</td><td>128 Hz (7.8125 ms)</td></tr> <tr><td>1010b</td><td>64 Hz (15.625 ms)</td></tr> <tr><td>1011b</td><td>32 Hz (31.25 ms)</td></tr> <tr><td>1100b</td><td>16 Hz (62.5 ms)</td></tr> <tr><td>1101b</td><td>8 Hz (125 ms)</td></tr> <tr><td>1110b</td><td>4 Hz (250 ms)</td></tr> <tr><td>1111b</td><td>2 Hz (500 ms)</td></tr> </tbody> </table>	Bits	Tap Frequency (Interrupt Rate)	0000b	Flat Signal (None)	0001b	256 Hz (3.90625 ms)	0010b	128 Hz (7.8125 ms)	0011b	8.192 kHz (122.070 us)	0100b	4.096 kHz (244.141 us)	0101b	2.048 kHz (488.281 us)	0110b	1.024 kHz (976.5625 us)	0111b	512 Hz (1.953125 ms)	1000b	256 Hz (3.90625 ms)	1001b	128 Hz (7.8125 ms)	1010b	64 Hz (15.625 ms)	1011b	32 Hz (31.25 ms)	1100b	16 Hz (62.5 ms)	1101b	8 Hz (125 ms)	1110b	4 Hz (250 ms)	1111b	2 Hz (500 ms)
Bits	Tap Frequency (Interrupt Rate)																																		
0000b	Flat Signal (None)																																		
0001b	256 Hz (3.90625 ms)																																		
0010b	128 Hz (7.8125 ms)																																		
0011b	8.192 kHz (122.070 us)																																		
0100b	4.096 kHz (244.141 us)																																		
0101b	2.048 kHz (488.281 us)																																		
0110b	1.024 kHz (976.5625 us)																																		
0111b	512 Hz (1.953125 ms)																																		
1000b	256 Hz (3.90625 ms)																																		
1001b	128 Hz (7.8125 ms)																																		
1010b	64 Hz (15.625 ms)																																		
1011b	32 Hz (31.25 ms)																																		
1100b	16 Hz (62.5 ms)																																		
1101b	8 Hz (125 ms)																																		
1110b	4 Hz (250 ms)																																		
1111b	2 Hz (500 ms)																																		

**IO073\_x0B RTC Register B**

Bits	Description
7	<b>SET: Set new time.</b> Read-write. Reset: 0. 1=No internal updating for RTC time registers occurs. 0=The RTC time registers are updated every second.
6	<b>PIE: Periodic Interrupt Enable.</b> Read-write. Reset: 0. 1=Enable the <a href="#">IO073_x0C[PF]</a> bit to assert IRQ.
5	<b>AIE: Alarm Interrupt Enable.</b> Read-write. Reset: 0. 1=Enable the <a href="#">IO073_x0C[AF]</a> bit to assert IRQ. When the alarm time is written in the appropriate hours, minutes, and seconds alarm registers, the alarm interrupt is initiated at the specified time each day if AIE=1.
4	<b>UIE: Update Ended Interrupt Enable.</b> Read-write. Reset: 0. 1=Enable the <a href="#">IO073_x0C[UF]</a> bit to assert IRQ. If [SET]=1, UIE is cleared.
3:2	Reserved.
1	<b>HourMode.</b> Read-write. Reset: 0. 0=12 hour mode. 1=24 hour mode.
0	<b>DaylightSavingEnable.</b> Read-write. Reset: X. BIOS: 0. 1=If ( <a href="#">PMx5F_x00[DltSavEnable]</a> ==1) RTC daylight saving is enabled.



**IO073\_x0C RTC Register C**

Bits	Description
7	<b>IRQF: Interrupt Request Flag.</b> Read-only. Reset: 0. $IRQF=(PF*PIE)+(AF*AIE)+(UF*UIE)$ . 1=The IRQ# pin is driven low. Reading RTC Register C clears IRQF bit.
6	<b>PF: Periodic Interrupt Flag.</b> Read-only. Reset: 0. 1=An edge is detected on the selected tap (through RS3 to RS0) of the frequency divider. Reading RTC Register C clears PF bit.
5	<b>AF: Alarm Interrupt Flag.</b> Read-only. Reset: 0. 1=IO073_x00 [RTC Seconds], IO073_x02 [RTC Minutes], IO073_x04 [RTC Hours] and IO073_x07 [RTC Date of Month] match IO073_x01 [RTC Seconds Alarm], IO073_x03 [RTC Minutes Alarm], IO073_x05 [RTC Hours Alarm], and IO073_x0D [RTC Date Alarm] respectively. Reading RTC Register C clears AF bit.
4	<b>UF: Update Ended Interrupt Flag.</b> Read-only. Reset: 0. 1=Update cycle complete. Reading RTC Register C clears UF.
3:0	Reserved.

**IO073\_x0D RTC Date Alarm**

Bits	Description
7	<b>VRT: Valid RAM and Time.</b> Read-only. Reset: 1. 1=RTC date, time, and CMOS RAM are valid. 0=RTC date, time, and CMOS RAM are invalid due to low RTC battery being monitored. See PMx58 [VRT_T1] and PMx59 [VRT_T2].
6	<b>ScratchBit.</b> Read-write. Reset: 0.
5:0	<b>DateAlarm.</b> Read-write. Reset: 0. If (DateAlarm!=0), then DateAlarm is considered for alarm generation. 0=DateAlarm is not compared for alarm generation. DateAlarm is in BCD format.

**IO073\_x32 RTC AltCentury**

Bits	Description
7:0	<b>AltCentury.</b> Read-write. Reset: 0. BCD format. Leap year correction is performed by hardware. This register is accessed only when (IO073_x0A[DV0]=0) and (PMx56[CenturyEn]=1). If (IO073_x0B[SET]=1), this register is programmed by software and hardware updating is disabled. If (IO073_x0B[SET]=0), this register is automatically updated by hardware every century.

**IO073\_x48 RTC Century**

Bits	Description
7:0	<b>Century.</b> Read-write. Reset: 0. BCD format. Leap year correction is done through hardware. This register is accessed only when (IO073_x0A[DV0]=1). If (IO073_x0B[SET]=1), this register is programmed by software and hardware updating is disabled. If (IO073_x0B[SET]=0), this register is automatically updated by hardware every century.

**IO073\_x50 RTC Extended RAM Address Port**

Bits	Description
7	Reserved.
6:0	<b>ExtendedRamAddrPort.</b> Read-write. Reset: 0. Because only 7 address bits are used in <a href="#">IO070[RtcAddrPort]</a> , only the lower 128 bytes at offset 7Fh:00h are accessible through <a href="#">IO071 [Rtc-DataPort]</a> . The Extended RAM are physically located at address 80h to FFh. In order to access these addresses, an address offset should be programmed into this register and access to them is done through <a href="#">IO073_x53 [RTC Extended RAM Data Port]</a> . An offset of 80h is automatically be added to this 7-bit address.

**IO073\_x53 RTC Extended RAM Data Port**

Bits	Description
7:0	<b>ExtendedRamDataPort.</b> Read-write. Reset: X.

**IO073\_x7E RTC Time Clear**

Bits	Description
7:1	Reserved.
0	<b>RtcTimeClear.</b> Read-write. Reset: 0. 1=Clear the RTC second and stop RTC time.

**IO073\_x7F RTC RAM Enable**

Bits	Description
7:1	Reserved.
0	<b>RtcRamEnable.</b> Read-write. Reset: 1. 1=Enable access to the RTC RAM.

**IO080 PostCode**

Bits	Description
7:0	<b>PostCode: BIOS post code.</b> Read-write. Reset: 0. BIOS post code register, can be 32-bits for writes, 8 bits for reads. See <a href="#">MISCx78 [PostCode]</a> .

**IO081 Dma\_PageCh2**

Bits	Description
7:0	<b>DmaPagech2.</b> Read-write. Reset: 0. DMA2 channel 2 page register.

**IO082 Dma\_PageCh3**

Bits	Description
7:0	<b>DmaPageCh3</b> . Read-write. Reset: 0. DMA2 channel 3 page register.

**IO083 Dma\_PageCh1**

Bits	Description
7:0	<b>DmaPageCh1</b> . Read-write. Reset: 0. DMA2 channel 1 page register.

**IO084 Dma\_Page\_Reserved1**

Bits	Description
7:0	<b>DmaPageReserved1</b> . Read-write. Reset: 0. DMA Page Reserved1 register.

**IO085 Dma\_Page\_Reserved2**

Bits	Description
7:0	<b>DmaPageReserved2</b> . Read-write. Reset: 0. DMA Page Reserved2 register.

**IO086 Dma\_Page\_Reserved3**

Bits	Description
7:0	<b>DmaPageReserved3</b> . Read-write. Reset: 0. DMA Page Reserved3 register.

**IO087 Dma\_PageCh0**

Bits	Description
7:0	<b>DmaPageCh0</b> . Read-write. Reset: 0. DMA2 channel 0 page register.

**IO088 Dma\_Page\_Reserved4**

Bits	Description
7:0	<b>DmaPageReserved4</b> . Read-write. Reset: 0. DMA Page Reserved4 register.

**IO089 Dma\_PageCh6**

Bits	Description
7:0	<b>DmaPageCh6</b> . Read-write. Reset: 0. DMA2 channel 6 page register.

**IO08A Dma\_PageCh7**

Bits	Description
7:0	<b>DmaPageCh7</b> . Read-write. Reset: 0. DMA2 channel 7 page register.

**IO08B Dma\_PageCh5**

Bits	Description
7:0	<b>DmaPageCh5</b> . Read-write. Reset: 0. DMA2 channel 5 page register.

**IO08C Dma\_Page\_Reserved5**

Bits	Description
7:0	<b>DmaPageReserved5</b> . Read-write. Reset: 0. DMA Page Reserved5 register.

**IO08D Dma\_Page\_Reserved6**

Bits	Description
7:0	<b>DmaPageReserved6</b> . Read-write. Reset: 0. DMA Page Reserved6 register.

**IO08E Dma\_Page\_Reserved7**

Bits	Description
7:0	<b>DmaPageReserved7</b> . Read-write. Reset: 0. DMA Page Reserved7 register.

**IO08F Dma\_Refresh**

Bits	Description
7:0	<b>DmaRefresh</b> . Read-write. Reset: 0. DMA2 channel 4 page register.

**IO092 FastInit**

Bits	Description
7:2	Reserved.
1	<b>A20EnB: A20 Enable Bar bit</b> . Read-write. Reset: 0. 1=A20M# function is disabled.
0	<b>FastInit</b> . Read-write. Reset: 0. This bit provides a fast software executed processor reset function. 1=Generate INIT assertion for approximately 4 ms. 0=Before another INIT pulse can be generated via this register, this bit must be written back to a 0.

**IO0A0 IntrCntrl2Reg1**

Bits	Description
7:0	<b>IntrCntrl2Reg1</b> . Read-write. Reset: 0. IRQ8-15 status and control. Read: IRR, ISR. Write: ICW1, OCW2, OCW3.

**IO0A1 IntrCntrl2Reg2**

Bits	Description
7:0	<b>IntrCntrl2Reg2</b> . Read-write. Reset: 0. IRQ8-15 status and control. Read: IMR. Write: ICW2, ICW3, ICW4, OCW1.

**IO0C0 Dma2\_Ch4Addr**

Bits	Description
7:0	<b>Dma2Ch4Addr</b> . Read-write. Reset: 0. DMA2 channel 4 base and current address.

**IO0C2 Dma2\_Ch4Cnt**

Bits	Description
7:0	<b>Dma2Ch4Cnt</b> . Read-write. Reset: 0. DMA2 channel 4 base and current count.

**IO0C4 Dma2\_Ch5Addr**

Bits	Description
7:0	<b>Dma2Ch5Addr</b> . Read-write. Reset: 0. DMA2 channel 5 base and current address.

**IO0C6 Dma2\_Ch5Cnt**

Bits	Description
7:0	<b>Dma2Ch5Cnt</b> . Read-write. Reset: 0. DMA2 channel 5 base and current count.

**IO0C8 Dma2\_Ch6Addr**

Bits	Description
7:0	<b>Dma2Ch6Addr</b> . Read-write. Reset: 0. DMA2 channel 6 base and current address.

**IO0CA Dma2\_Ch6Cnt**

Bits	Description
7:0	<b>Dma2Ch6Cnt.</b> Read-write. Reset: 0. DMA2 channel 6 base and current count.

**IO0CC Dma2\_Ch7Addr**

Bits	Description
7:0	<b>Dma2Ch7Addr.</b> Read-write. Reset: 0. DMA2 channel 7 base and current address.

**IO0CE Dma\_Ch7Cnt**

Bits	Description
7:0	<b>Dma2Ch7Cnt.</b> Read-write. Reset: 0. DMA2 channel 7 base and current count.

**IO0D0 Dma\_Status**

Bits	Description
7:0	<b>DmaStatus.</b> Read-write. Reset: 0. DMA2 status register.

**IO0D2 Dma\_WriteRequest**

Bits	Description
7:0	<b>DmaWriteRequest.</b> Read-write. Reset: 0. DMA2 request register.

**IO0D4 Dma\_WriteMask**

Bits	Description
7:0	<b>DmaWriteMask.</b> Read-write. Reset: 0. DMA2 channel mask register.

**IO0D6 Dma\_WriteMode**

Bits	Description
7:0	<b>DmaWriteMode.</b> Read-write. Reset: 0. DMA2 mode register.

**IO0D8 Dma\_Clear**

Bits	Description
7:0	<b>DmaClear.</b> Read-write. Reset: 0. Channel 4-7 clear byte pointer.

**IO0DA Dma\_MasterClr**

Bits	Description
7:0	<b>DmaMasterClear</b> . Read-write. Reset: 0. Write: Channel 4-7 DMA master clear. Read: Intermediate register.

**IO0DC Dma\_ClrMask**

Bits	Description
7:0	<b>DmaClrMask</b> . Read-write. Reset: 0. Channel 4-7 DMA clear mask.

**IO0DE Dma\_AllMask**

Bits	Description
7:0	<b>DmaAllMask</b> . Read-write. Reset: 0. DMA2 mask register.

**IO0F0 NCP Error**

Writes to this port assert IGNNE# if FERR# is true. If FERR# is false, writes to this port do not assert IGNNE#. The first write will set WarmBoot.

Bits	Description
7	<b>WarmBoot</b> . Read; write-once. Cold reset: 0. Warm or cold boot indicator. 0=Cold. 1=Warm.
6:0	Reserved.

**IO4D0 IntrEdgeControl**

Bits	Description
15	<b>IRQ15Control</b> . Read-write. Reset: 0. 1=Level. 0=Edge.
14	<b>IRQ14Control</b> . Read-write. Reset: 0. 1=Level. 0=Edge.
13	Reserved.
12	<b>IRQ12Control</b> . Read-write. Reset: 0. 1=Level. 0=Edge.
11	<b>IRQ11Control</b> . Read-write. Reset: 0. 1=Level. 0=Edge.
10	<b>IRQ10Control</b> . Read-write. Reset: 0. 1=Level. 0=Edge.
9	<b>IRQ9Control</b> . Read-write. Reset: 0. 1=Level. 0=Edge.
8	<b>IRQ8Control</b> . Read-only. Reset: 0. Always edge.
7	<b>IRQ7Control</b> . Read-write. Reset: 0. 1=Level. 0=Edge.
6	<b>IRQ6Control</b> . Read-write. Reset: 0. 1=Level. 0=Edge.
5	<b>IRQ5Control</b> . Read-write. Reset: 0. 1=Level. 0=Edge.
4	<b>IRQ4Control</b> . Read-write. Reset: 0. 1=Level. 0=Edge.
3	<b>IRQ3Control</b> . Read-write. Reset: 0. 1=Level. 0=Edge.

2	Reserved.
1	<b>IRQ1Control.</b> Read-write. Reset: 0. 1=Level. 0=Edge.
0	<b>IRQ0Control.</b> Read-write. Reset: 0. 1=Level. 0=Edge.

**IOC00 Pci\_Intr\_Index**

Bits	Description																																																																																																				
7	<b>PciIntrApic.</b> Read-write. Reset: 0. 0=IRQ routing to PIC. 1=IRQ routing to IOAPIC.																																																																																																				
6:0	<b>PciIntrIndex.</b> Read-write. Reset: 0. PCI interrupt index. Selects which PCI interrupt to map.																																																																																																				
	<table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00h</td> <td>INTA#</td> <td>18h</td> <td>Reserved</td> </tr> <tr> <td>01h</td> <td>INTB#</td> <td>19h</td> <td>Reserved</td> </tr> <tr> <td>02h</td> <td>INTC#</td> <td>1Fh-1Ah</td> <td>Reserved</td> </tr> <tr> <td>03h</td> <td>INTD#</td> <td>20h</td> <td>IMC INT0</td> </tr> <tr> <td>04h</td> <td>GENINT1_L</td> <td>21h</td> <td>IMC INT1</td> </tr> <tr> <td>05h</td> <td>GENINT2_L</td> <td>22h</td> <td>IMC INT2</td> </tr> <tr> <td>06h</td> <td>Reserved</td> <td>23h</td> <td>IMC INT3</td> </tr> <tr> <td>07h</td> <td>Reserved</td> <td>24h</td> <td>IMC INT4</td> </tr> <tr> <td>08h</td> <td>Misc</td> <td>25h</td> <td>IMC INT5</td> </tr> <tr> <td>09h</td> <td>Misc0</td> <td>2Fh-26h</td> <td>Reserved</td> </tr> <tr> <td>0Ah</td> <td>Misc1</td> <td>30h</td> <td>Dev12h (USB) INTA#</td> </tr> <tr> <td>0Bh</td> <td>Misc2</td> <td>31h</td> <td>Dev12h (USB) INTB#</td> </tr> <tr> <td>0Ch</td> <td>INTA from serial IRQ</td> <td>32h</td> <td>Dev13h (USB) INTA#</td> </tr> <tr> <td>0Dh</td> <td>INTB from serial IRQ</td> <td>33h</td> <td>Dev13h (USB) INTB#</td> </tr> <tr> <td>0Eh</td> <td>INTC from serial IRQ</td> <td>34h</td> <td>Dev16h (USB) INTA#</td> </tr> <tr> <td>0Fh</td> <td>INTD from serial IRQ</td> <td>35h</td> <td>Dev16h (USB) INTB#</td> </tr> <tr> <td>10h</td> <td>SCI</td> <td>40h-36h</td> <td>Reserved</td> </tr> <tr> <td>11h</td> <td>SMBUS0</td> <td>41h</td> <td>SATA PCI interrupt</td> </tr> <tr> <td>12h</td> <td>ASF</td> <td>7Fh-42h</td> <td>Reserved</td> </tr> <tr> <td>13h</td> <td>HD audio</td> <td></td> <td></td> </tr> <tr> <td>14h</td> <td>FC</td> <td></td> <td></td> </tr> <tr> <td>15h</td> <td>Reserved</td> <td></td> <td></td> </tr> <tr> <td>16h</td> <td>PerMon</td> <td></td> <td></td> </tr> <tr> <td>17h</td> <td>SD</td> <td></td> <td></td> </tr> </tbody> </table>	Bits	Definition	Bits	Definition	00h	INTA#	18h	Reserved	01h	INTB#	19h	Reserved	02h	INTC#	1Fh-1Ah	Reserved	03h	INTD#	20h	IMC INT0	04h	GENINT1_L	21h	IMC INT1	05h	GENINT2_L	22h	IMC INT2	06h	Reserved	23h	IMC INT3	07h	Reserved	24h	IMC INT4	08h	Misc	25h	IMC INT5	09h	Misc0	2Fh-26h	Reserved	0Ah	Misc1	30h	Dev12h (USB) INTA#	0Bh	Misc2	31h	Dev12h (USB) INTB#	0Ch	INTA from serial IRQ	32h	Dev13h (USB) INTA#	0Dh	INTB from serial IRQ	33h	Dev13h (USB) INTB#	0Eh	INTC from serial IRQ	34h	Dev16h (USB) INTA#	0Fh	INTD from serial IRQ	35h	Dev16h (USB) INTB#	10h	SCI	40h-36h	Reserved	11h	SMBUS0	41h	SATA PCI interrupt	12h	ASF	7Fh-42h	Reserved	13h	HD audio			14h	FC			15h	Reserved			16h	PerMon			17h	SD		
Bits	Definition	Bits	Definition																																																																																																		
00h	INTA#	18h	Reserved																																																																																																		
01h	INTB#	19h	Reserved																																																																																																		
02h	INTC#	1Fh-1Ah	Reserved																																																																																																		
03h	INTD#	20h	IMC INT0																																																																																																		
04h	GENINT1_L	21h	IMC INT1																																																																																																		
05h	GENINT2_L	22h	IMC INT2																																																																																																		
06h	Reserved	23h	IMC INT3																																																																																																		
07h	Reserved	24h	IMC INT4																																																																																																		
08h	Misc	25h	IMC INT5																																																																																																		
09h	Misc0	2Fh-26h	Reserved																																																																																																		
0Ah	Misc1	30h	Dev12h (USB) INTA#																																																																																																		
0Bh	Misc2	31h	Dev12h (USB) INTB#																																																																																																		
0Ch	INTA from serial IRQ	32h	Dev13h (USB) INTA#																																																																																																		
0Dh	INTB from serial IRQ	33h	Dev13h (USB) INTB#																																																																																																		
0Eh	INTC from serial IRQ	34h	Dev16h (USB) INTA#																																																																																																		
0Fh	INTD from serial IRQ	35h	Dev16h (USB) INTB#																																																																																																		
10h	SCI	40h-36h	Reserved																																																																																																		
11h	SMBUS0	41h	SATA PCI interrupt																																																																																																		
12h	ASF	7Fh-42h	Reserved																																																																																																		
13h	HD audio																																																																																																				
14h	FC																																																																																																				
15h	Reserved																																																																																																				
16h	PerMon																																																																																																				
17h	SD																																																																																																				

**IOC01 Pci\_Intr\_Data**

Bits	Description
7:0	<b>PciIntrData.</b> Read-write. Reset: 0.

**IOC01\_x0[7:0] PCI INT[H#,G#,F#,E#,D#,C#,B#,A#] Map**

Bits	Description
------	-------------



7:5	Reserved.
4:0	<b>Pci2IntrMap</b> . Read-write. Reset: 1Fh. If (IOC00[PciIntrApic]==1) then Pci2IntrMap specifies mapping of INT[H#:A#] to APIC interrupt number. If (IOC00[PciIntrApic]==0) then Pci2IntrMap specifies mapping of INT[H#:A#] to PIC interrupt number.

**IOC01\_x08 Intr Misc Map**

Bits	Description										
7:6	<b>Pci2Intr15Map</b> . Read-write. Reset: 0. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>IRQ15 mapped to legacy IDE</td> </tr> <tr> <td>01b</td> <td>IRQ15 mapped to SATA IDE</td> </tr> <tr> <td>10b</td> <td>IRQ15 mapped to SATA2</td> </tr> <tr> <td>11b</td> <td>IRQ15 come from Serial IRQ or PCI interrupt</td> </tr> </tbody> </table>	Bits	Definition	00b	IRQ15 mapped to legacy IDE	01b	IRQ15 mapped to SATA IDE	10b	IRQ15 mapped to SATA2	11b	IRQ15 come from Serial IRQ or PCI interrupt
Bits	Definition										
00b	IRQ15 mapped to legacy IDE										
01b	IRQ15 mapped to SATA IDE										
10b	IRQ15 mapped to SATA2										
11b	IRQ15 come from Serial IRQ or PCI interrupt										
5:4	<b>Pci2Intr14Map</b> . Read-write. Reset: 0. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>IRQ14 mapped to legacy IDE</td> </tr> <tr> <td>01b</td> <td>IRQ14 mapped to SATA IDE</td> </tr> <tr> <td>10b</td> <td>IRQ14 mapped to SATA2</td> </tr> <tr> <td>11b</td> <td>IRQ14 mapped to Serial IRQ or PCI interrupt</td> </tr> </tbody> </table>	Bits	Definition	00b	IRQ14 mapped to legacy IDE	01b	IRQ14 mapped to SATA IDE	10b	IRQ14 mapped to SATA2	11b	IRQ14 mapped to Serial IRQ or PCI interrupt
Bits	Definition										
00b	IRQ14 mapped to legacy IDE										
01b	IRQ14 mapped to SATA IDE										
10b	IRQ14 mapped to SATA2										
11b	IRQ14 mapped to Serial IRQ or PCI interrupt										
3	<b>PciIntrIrq12</b> . Read-write. Reset: 0. 0=IMC as IRQ12 input source. 1=Serial IRQ or PCI devices as IRQ12 input source.										
2	<b>PciIntrIrq8</b> . Read-write. Reset: 0. 0=RTC is IRQ8 input source. 1=Serial IRQ or PCI devices as IRQ8 input source.										
1	<b>PciIntrIrq1</b> . Read-write. Reset: 0. 0=IMC as IRQ1 input source. 1=Serial IRQ or PCI devices as IRQ1 input source.										
0	<b>PciIntrIrq0</b> . Read-write. Reset: 0. 0=8254 timer as IRQ0 input source. 1=Serial IRQ or PCI devices as IRQ0 input source.										

**IOC01\_x09 Intr Misc 0 Map**

Bits	Description
7	<b>IntrDelay</b> . Read-write. Reset: 1. INTR 600 ns delay.
6	<b>IRQ12FilterEnable</b> . Read-write. Reset: 1. IRQ12 filter enable.
5	<b>IRQ1FilterEnable</b> . Read-write. Reset: 1. IRQ1 filter enable.
4	<b>IrqInputEn</b> . Read-write. Reset: 0. 0=Mask off IRQ input. 1=Enable IRQ input.
3	<b>MaskIrq1Irq12</b> . Read-write. Reset: 0. 0=Enable IRQ1 and IRQ12. 1=Mask off IRQ1 and IRQ12.
2	<b>Merge_Ec_irq12</b> . Read-write. Reset: 1. 0=Route serial IRQ12 to USB IRQ12 input. 1=Route IMC IRQ12 to USB IRQ12 input.
1	<b>Merge_Ec_irq1</b> . Read-write. Reset: 1. 0=Route serial IRQ1 to USB IRQ1 input. 1=Route IMC IRQ1 to USB IRQ1 input.
0	<b>IntMap</b> . Read-write. Reset: 1. 0=INT0 in IOAPIC comes from IRQ0 in PIC, INT2 in IOAPIC comes from INTR in PIC. 1=INT2 in IOAPIC comes from IRQ0 in PIC, INT0 in IOAPIC comes from INTR in PIC.

**IOC01\_x0A IntrMisc1Map**

Bits	Description
7:0	<b>HPET</b> . Read-write. Reset: 0. Writes to this register update the bits in <a href="#">HPETx1[4:0:Step2]0[47:32]</a> ; All 3 registers, <a href="#">HPETx100[47:32]</a> , <a href="#">HPETx120[47:32]</a> , and <a href="#">HPETx140[47:32]</a> , are updated at the same time. <a href="#">IOC01_x0A</a> updates the lower 8 bits. <a href="#">IOC01_x0B</a> updates the upper 8 bits.

**IOC01\_x0B IntrMisc2Map**

Bits	Description
7:0	<b>HPET</b> . See: <a href="#">IOC01_x0A[HPET]</a> .

**IOC01\_x0[7F:C] PCI Interrupt Map**

Bits	Description
7:5	Reserved.
4:0	<b>Pci2IntrMap</b> . Read-write. Reset: 1Fh. If ( <a href="#">IOC00[PciIntrApic]</a> ==1), then <a href="#">Pci2IntrMap</a> specifies the APIC interrupt number that the corresponding PCI interrupt maps to. If ( <a href="#">IOC00[PciIntrApic]</a> ==0), then <a href="#">Pci2IntrMap</a> specifies the PIC interrupt number that the corresponding PCI interrupt maps to. See <a href="#">IOC00 [Pci_Intr_Index]</a> for the PCI interrupt list.

**IOC14 Pci\_Error**

Bits	Description
7:4	Reserved.
3	<b>PerrNmi</b> . Read-write. Reset: 1. Enable NMI generation from PERR#. 0=Enable. 1=Disable.
2	<b>SerrNmi</b> . Read-write. Reset: 1. Enable NMI generation from SERR#. 0=Enable. 1=Disable.
1	<b>PerrNmiStatus</b> . Read-only. Reset: X. 1=NMI generation is enabled and PERR# is asserted due to a PCI data parity error. This bit is cleared by writing 1 to <a href="#">IO061[2]</a> .
0	<b>SerrNmiStatus</b> . Read-only. Reset: X. 1=NMI generation is enabled and SERR# is asserted due to a PCI error. This bit is cleared by writing 1 to <a href="#">IO061[2]</a> .

**IOCD0 PM2\_Index**

Bits	Description
7:0	<b>Pm2Index: Power management 2 index register</b> . Read-write. Reset: 0. This register specifies the index of the power management 2 register. See <a href="#">Power Management Block 2 (PM2) Registers</a> .

**IOCD1 PM2\_Data**

Bits	Description
7:0	<b>Pm2Data: Power management 2 data register.</b> Read-write. Reset: 0. This register specifies the data read from/written to the power management 2 register pointed by <a href="#">IOCD0</a> . See <a href="#">Power Management Block 2 (PM2) Registers</a> .

**IOCD4 BIO SRAMIndex**

Bits	Description
7:0	<b>BiosRamIndex: BIOS RAM index register.</b> Read-write. Reset: 0. This register specifies the index in the 256-byte BIOS RAM. Data in this RAM is preserved until RSMRST# is asserted or S5 power is lost.

**IOCD5 BIO SRAM Data**

Bits	Description
7:0	<b>BiosRamData: BIOS RAM data register.</b> Read-write. Reset: 0. This register specifies the data read from/written to the BIOS RAM pointed by <a href="#">IOCD4</a> .

**IOCD6 PM\_Index**

Bits	Description
7:0	<b>PmIndex: Power management index register.</b> Read-write. Reset: 0. This register specifies the index of the power management register. See <a href="#">3.25.13 [Power Management (PM) Registers]</a> .

**IOCD7 PM\_Data**

Bits	Description
7:0	<b>PmData: Power management data register.</b> Read-write. Reset: 0. This register specifies the data read from/written to the power management register pointed by <a href="#">IOCD6</a> . See <a href="#">3.25.13 [Power Management (PM) Registers]</a> .

**IOCF9 System Reset Register**

This register can be accessed through [PMxC5 \[CF9 Shadow\]](#).

Bits	Description
7:4	Reserved.
3	<b>FullRst.</b> Read-write. Reset: 0. 0=Assert reset signals only. 1=Place system in S5 state for 3 to 5 seconds.
2	<b>RstCmd.</b> Read-write; Cleared-by-hardware. Reset: 0. 1=Generate reset as specified by FullRst and SysRst.

1	<b>SysRst.</b> Read-write. Reset: 0. 0=Send an INIT HT message. 1=Reset as specified by FullRst.
0	Reserved.

### 3.25.2 AB Configuration Registers

AB Configuration registers are accessed indirectly through [ABx00 \[AB Index Register\]](#) and [ABx04 \[AB Data Register\]](#). The [ABx00/ABx04](#) register pair is located in IO address space as defined by [PMxE0](#).

#### ABx00 AB Index Register

Bits	Description								
31:29	<p><b>RegSpace.</b> Read-write. Reset: 0.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>101b-000b</td> <td>Reserved</td> </tr> <tr> <td>110b</td> <td>ABCFG. A-Link Bridge Configuration</td> </tr> <tr> <td>111b</td> <td>Reserved</td> </tr> </tbody> </table> <p>Once a valid register block address is written to this field, only valid register block addresses can be written to this field.</p>	Bits	Definition	101b-000b	Reserved	110b	ABCFG. A-Link Bridge Configuration	111b	Reserved
Bits	Definition								
101b-000b	Reserved								
110b	ABCFG. A-Link Bridge Configuration								
111b	Reserved								
28:17	Reserved.								
16:0	<b>RegAddr: Register Address.</b> Read-write. Reset: 0.								

#### ABx04 AB Data Register

Bits	Description
31:0	<b>Data: Register Address.</b> Read-write. Reset: 0.

#### ABx04\_x54 Misc Control 1

Bits	Description
31:25	Reserved.
24	<b>BIClkGateEn.</b> Read-write. Reset: 0. BIOS: See <a href="#">2.15.11.3</a> . 1=Enable gating of B-Link clocks when idle is detected.
23:16	<b>BIClkGateDelay.</b> Read-write. Reset: 0. BIOS: See <a href="#">2.15.11.3</a> . Specifies the number of B-Link clocks to delay before gating B-Link clocks after idle condition is detected.
15:5	Reserved.
4	<b>DbgClkGateEn.</b> Read-write. Reset: 0. BIOS: See <a href="#">2.15.11.3</a> . 1=Enable gating of B-Link debug clocks.
3	Reserved.
2	<b>UpSWrByteCntSbgMode.</b> Read-write. Reset: 0. BIOS: See <a href="#">2.15.11.4</a> . 1=Upstream 32/64 byte MST_BIF write requests will be issued only when all byte enables in the request are 1. This field is mutually exclusive with UpWr16BMode.
1	Reserved.
0	<b>UpWr16BMode.</b> Read-write. Reset: 1. BIOS: See <a href="#">2.15.11.4</a> . Enables upstream writes to be limited to a maximum of 16-bytes per transaction. 0=Disable 16 Byte mode; AB can send DMA write request with length 16, 32, or 64 bytes. 1=Enable 16 Byte mode; AB will only send DMA write request with length 16 bytes. This bit should be mutually exclusive with UpSWrByteCntSbgMode. See <a href="#">ABx04_x204[Dma16ByteMode]</a> .

**ABx04\_x58 B-Link RAB Control**

Bits	Description
31	<b>AlMemSDEn.</b> Read-write. Reset: 0. BIOS: See 2.15.11.2. 1=Enable AB A-Link Memory Shut Down Feature. See ABx04_x10054[27:26] for idle counter settings.
30	Reserved.
29	<b>BlMemSDEn.</b> Read-write. Reset: 0. BIOS: See 2.15.11.2. 1=Enable AB B-Link Memory Shut Down Feature. See ABx04_x10054[27:26] for idle counter settings.
28:0	Reserved.

**ABx04\_x80 B-Link DMA Prefetch Control**

Bits	Description										
31:22	Reserved.										
21:16	<b>BiPrefMode.</b> Read-write. Reset: 0. BIOS: 6. <table border="1"> <thead> <tr> <th>Bit</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>[5:3]</td> <td>Reserved.</td> </tr> <tr> <td>[2]</td> <td>Reserved, must be programmed to 1 to set B-Link prefetch mode</td> </tr> <tr> <td>[1]</td> <td>Reserved, must be programmed to 1 to set B-Link prefetch mode</td> </tr> <tr> <td>[0]</td> <td>Reserved.</td> </tr> </tbody> </table>	Bit	Definition	[5:3]	Reserved.	[2]	Reserved, must be programmed to 1 to set B-Link prefetch mode	[1]	Reserved, must be programmed to 1 to set B-Link prefetch mode	[0]	Reserved.
Bit	Definition										
[5:3]	Reserved.										
[2]	Reserved, must be programmed to 1 to set B-Link prefetch mode										
[1]	Reserved, must be programmed to 1 to set B-Link prefetch mode										
[0]	Reserved.										
15:8	Reserved.										
7:0	<b>BiPrefEn.</b> Read-write. Reset: 0. BIOS: 1. Enable B-Link prefetch on a per-device basis. <table border="1"> <thead> <tr> <th>Bit</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>[7:1]</td> <td>Reserved.</td> </tr> <tr> <td>[0]</td> <td>USB OHCI.</td> </tr> </tbody> </table>	Bit	Definition	[7:1]	Reserved.	[0]	USB OHCI.				
Bit	Definition										
[7:1]	Reserved.										
[0]	USB OHCI.										

**ABx04\_x90 BIF Control 0**

Bits	Description
31:22	Reserved.
21	<b>PpNpRfwCplCtrl.</b> Read-write. Reset: 0. BIOS: 1. RFW completion control for posted pass non-posted.
20:0	Reserved.

**ABx04\_x94 MSI Control**

Bits	Description
31:21	Reserved.
20	<b>MsiAddrEn.</b> Read-write. Reset: 0. BIOS: 1. 1=Enable AB to detect MSI sent upstream from FCH and inform power-management controller.
19:0	<b>MsiAddr[39:20].</b> Read-write. Reset: 0. BIOS: See 2.15.11.1. Specifies MSI Address[39:20].

**ABx04\_x204 SBG Upstream Control**

Bits	Description
31:1	Reserved.
0	<b>Dma16ByteMode</b> . Read-write. Reset: 0. To always split DMA write request to be 16 bytes request. 0=All the DMA write requests has the size that AB passes up. 1=Split the DMA write requests to be 16 bytes requests. Overrides <a href="#">ABx04_x54[UpWr16BMode]</a> .

**ABx04\_x10054 A-Link Arbitration Control and Clock Control**

Bits	Description
31:26	Reserved.
25	<b>DbgClkGateEn</b> . Read-write. Reset: 0. BIOS: See <a href="#">2.15.11.3</a> . 1=Enable gating of A-Link debug clocks.
24	<b>AIclkGateEn</b> . Read-write. Reset: 0. BIOS: See <a href="#">2.15.11.3</a> . 1=Enable gating of A-Link clocks when idle is detected.
23:16	<b>AIclkGateDelay</b> . Read-write. Reset: 0. BIOS: See <a href="#">2.15.11.3</a> . Specifies the number of cycles to delay before gating A-Link clocks after idle condition is detected.
15:12	Reserved.
11:0	<b>ArbCtl</b> . Read-write. Reset: 0. BIOS: 7FFh. A-Link Arbiter Control. Enable the A-Link int_arbiter enhancement to allow the A-Link bandwidth to be used more efficiently

**ABx04\_x10090 Misc Control 3**

Bits	Description
31:17	Reserved.
16	<b>IOTrapDelayEnable: IO Trap Delay Enable</b> . Read-write. Reset: 0. BIOS: 1. 1=Enable the IO trap delay.
15:0	Reserved.

### 3.25.3 SATA Controller

#### 3.25.3.1 Device 11h Function 0 (SATA) 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.

#### D11F0x00 Device/Vendor ID

Bits	Description								
31:16	<p><b>DeviceID:</b> Read-only. Reset: Product-specific. SATA controller has different device IDs for different drivers.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>7800h</td> <td>SATA in IDE mode</td> </tr> <tr> <td>7801h</td> <td>SATA in AHCI mode with MS driver</td> </tr> <tr> <td>7804h</td> <td>SATA in AHCI mode with AMD driver</td> </tr> </tbody> </table>	Bits	Definition	7800h	SATA in IDE mode	7801h	SATA in AHCI mode with MS driver	7804h	SATA in AHCI mode with AMD driver
Bits	Definition								
7800h	SATA in IDE mode								
7801h	SATA in AHCI mode with MS driver								
7804h	SATA in AHCI mode with AMD driver								
15:0	<b>VendorID: Vendor ID.</b> Read-only. Reset: 1022h.								

#### D11F0x04 Status/Command

Bits	Description
31	<b>DetectedParityError: Detected Parity Error.</b> Read; write-1-to-clear. Reset: 0. 1=The SATA controller detects a parity error.
30	<b>SerrStatus: SERR# Status.</b> Read; write-1-to-clear. Reset: 0. 1=The SATA controller detects a PCI address parity error.
29	<b>ReceivedMasterAbort: Received Master Abort.</b> Read; write-1-to-clear. Reset: 0. 1=The SATA controller aborts a PCI bus memory cycle while acting as a PCI master.
28	<b>ReceivedTargetAbort: Received Target Abort.</b> Read; write-1-to-clear. Reset: 0. 1=The SATA controller generated PCI cycle (SATA controller is the PCI master) is aborted by a PCI target.
27	<b>SignaledTargetAbort: Signaled Target Abort.</b> Read; write-1-to-clear. Reset: 0. 1=The SATA controller signals Target Abort.
26:25	<b>DevselTiming: DEVSEL# Timing.</b> Value: 1. These bits indicate DEVSEL# timing when performing a positive decode. 1=DEVSEL# is asserted to meet the medium timing.
24	<b>DataParityError: Data Parity Error.</b> Read; write-1-to-clear. Reset: 0. 1=SATA controller detects PERR# being asserted while acting as PCI master, regardless whether PERR# was driven by SATA controller or not.
23	<b>FastBack2BackCapable: Fast Back-to-Back Capable.</b> Value: 0.
22	Reserved.
21	<b>Support66MHz: 66 MHz Support.</b> Read-only. Reset: 1. 1=66 MHz capable.
20	<b>CapabilitiesList: Capabilities List.</b> Read-only. Reset: 1. 1=Capability list supported.
19	<b>InterruptStatus: Interrupt Status.</b> Read-only. Reset: 0. This bit reflects the state of the interrupt in the device/function. If (InterruptDisable==0) && (InterruptStatus==1), the device/function's INTx# signal is asserted. Setting InterruptDisable to 1 has no effect on the state of this bit.



18:11	Reserved.
10	<b>InterruptDisable: Interrupt Disable.</b> Read-write. Reset: 0. This bit disables the device/function from asserting INTx#. 0=Enable the assertion of the device/function's INTx# signal. 1=Disable the assertion of the device/function's INTx# signal.
9	<b>FastBack2BackEnable: Fast Back-to-Back Enable.</b> Value: 0.
8	<b>SERREnable: SERR# Enable.</b> Read-write. Reset: 0. 1=The SATA controller asserts SERR# when it detects an address parity error && (PerrDetectionEnable==1). 0=SERR# is not asserted.
7	<b>WaitCycleEnable: Wait Cycle Enable.</b> Value: 0.
6	<b>PerrDetectionEnable: PERR# Detection Enable.</b> Read-write. Reset: 0. 1=The host controller asserts PERR# when it is the agent receiving data && when it detects a parity error. 0=PERR# is not asserted.
5	<b>VGAPaletteSnoopEnable: VGA Palette Snoop Enable.</b> Value: 0.
4	<b>MemoryWriteandInvalidateEnable: Memory Write and Invalidate Enable.</b> Value: 0.
3	<b>SpecialCycleRecognitionEnable: Special Cycle Recognition Enable.</b> Value: 0.
2	<b>BusMasterEnable: Bus Master Enable.</b> Read-write. Reset: 0. 1=Enable the device behaving as a bus master.
1	<b>MemoryAccessEnable: Memory Access Enable.</b> Read-write. Reset: 0. 1=Enable the SATA controller to respond to PCI memory space access.
0	<b>IOAccessEnable: IO Access Enable.</b> Read-write. Reset: 0. 1=Enable the SATA controller to respond to PCI IO space access.

**D11F0x08 Revision ID/Class Code**

SATA controller has different values for Subclass Code and ProgramIF[7:0] depending on its operating mode.

**Table 231: SATA Controller Subclass Code and ProgramIF Settings**

Subclass Code	ProgramIF[7:0]	Controller Type
01	8Fh/8Ah	IDE
06	01	AHCI

Bits	Description
31:24	<b>ClassCode: Class Code.</b> Value: 1. This is a Mass-Storage Controller.
23:16	<b>SubclassCode.</b> IF (D11F0x40[SubclassCodeWriteEnable]) THEN Read-write. ELSE Read-only. ENDIF. Reset: 01h. BIOS: See 2.15.3.1. See Table 231.
15	<b>ProgramIF[7]: Master IDE Device.</b> IF (D11F0x40[SubclassCodeWriteEnable]) THEN Read-write. ELSE Read-only. ENDIF. Reset: 1. BIOS: See 2.15.3.1. 1=Master IDE device in IDE mode. See Table 231.
14:12	Reserved.
11	<b>ProgramIF[3]: Secondary Operating Mode Programmable.</b> IF ((D11F0x40[SubclassCodeWriteEnable])    (D11F0x08[SubclassCode] == 01h)) THEN Read-write. ELSE Read-only. ENDIF. Reset: 1. BIOS: See 2.15.3.1. 1=Both secondary operating modes are supported in IDE. See Table 231

10	<b>ProgramIF[2]: Secondary Operating Mode.</b> IF ((D11F0x40[SubclassCodeWriteEnable] == 1)    (D11F0x08[SubclassCode] == 01h)) THEN Read-write. ELSE Read-only. ENDIF. Reset: 1. BIOS: See 2.15.3.1. See Table 231 <table><thead><tr><th><u>Bits</u></th><th><u>Definition</u></th></tr></thead><tbody><tr><td>0b</td><td>Compatibility Mode in IDE</td></tr><tr><td>1b</td><td>Native PCI Mode in IDE</td></tr></tbody></table>	<u>Bits</u>	<u>Definition</u>	0b	Compatibility Mode in IDE	1b	Native PCI Mode in IDE
<u>Bits</u>	<u>Definition</u>						
0b	Compatibility Mode in IDE						
1b	Native PCI Mode in IDE						
9	<b>ProgramIF[1]: Primary Operating Mode Programmable.</b> IF ((D11F0x40[SubclassCodeWriteEnable] == 1)    (D11F0x08[SubclassCode] == 01h)) THEN Read-write. ELSE Read-only. ENDIF. Reset: 1. BIOS: See 2.15.3.1. 1=Both Primary operating modes are supported in IDE mode. See Table 231						
8	<b>ProgramIF[0]: Primary Operating Mode.</b> IF ((D11F0x40[SubclassCodeWriteEnable]==1)    (D11F0x08[SubclassCode] == 01h)) THEN Read-write. ELSE Read-only. ENDIF. Reset: 1. BIOS: See 2.15.3.1. See Table 231. <table><thead><tr><th><u>Bits</u></th><th><u>Definition</u></th></tr></thead><tbody><tr><td>0b</td><td>Compatibility Mode in IDE</td></tr><tr><td>1b</td><td>Native PCI Mode in IDE</td></tr></tbody></table>	<u>Bits</u>	<u>Definition</u>	0b	Compatibility Mode in IDE	1b	Native PCI Mode in IDE
<u>Bits</u>	<u>Definition</u>						
0b	Compatibility Mode in IDE						
1b	Native PCI Mode in IDE						
7:0	<b>RevisionID: Revision ID.</b> Read-only. Reset: 39h.						

**D11F0x0C Header Type Register**

Bits	Description
31	<b>BISTCapable: BIST Capable.</b> Value: 0. 0=No HBA related BIST function.
30	<b>StartBIST: Start BIST.</b> Read-write. Reset: 0. Programming this bit has no effect.
29:28	Reserved.
27:24	<b>CompletionCode: Completion Code.</b> Read-only. Reset: 0. Indicates the completion code status of BIST. A non-zero value indicates a failure.
23	<b>MultiFunctionDevice: Multi-function device.</b> Read-only. Reset: 0. 0=Single-function device.
22:16	<b>HeaderType: Header Type.</b> Read-only. Reset: 0. Since the SATA host controller is a single-function device, this field contains a value of 00h.
15:11	<b>MasterLatencyTimer: Master Latency Timer.</b> Read-write. Reset: 0. Specifies, in units of PCI bus clocks, the time slice allowed to SATA host controller for burst transactions.
10:8	Reserved.
7:4	<b>CacheLineSize: Cache Line Size.</b> Read-write. Reset: 0.
3:0	Reserved.

**D11F0x10 Primary IDE CS0 Base Address(BAR0)**

Bits	Description
31:3	<b>PrimaryIDECS0BaseAddress: Primary IDE CS0 Base Address.</b> Read-write. Reset: 0. Base address[31:3] for Primary IDE Bus CS0 in IDE mode. This register is used for native mode only. See 3.25.3.2.1 [IDE Compatibility Mode and Native Mode Registers]
2:1	Reserved.
0	<b>ResourceTypeIndicator: Resource Type Indicator.</b> Value: 1. 1=Base address field in this register maps to the IO space.

**D11F0x14 Primary IDE CS1 Base Address (BAR1)**

Bits	Description
31:2	<b>PrimaryIDECS1BaseAddress: Primary IDE CS1 Base Address.</b> Read-write. Reset: 0. Base address[31:2] for Primary IDE Bus CS1 in IDE mode. This register is used for native mode only. See <a href="#">3.25.3.2.1 [IDE Compatibility Mode and Native Mode Registers]</a>
1	Reserved.
0	<b>ResourceTypeIndicator: Resource Type Indicator.</b> Value: 1. 1=Base address field in this register maps to the IO space.

**D11F0x18 Secondary IDE CS0 Base Address (BAR2)**

Bits	Description
31:3	<b>SecondaryIDECS0BaseAddress: Secondary IDE CS0 Base Address.</b> Read-write. Reset: 0. Base address[31:3] for Secondary IDE Bus CS0 in IDE mode. This register is used for native mode only. See <a href="#">3.25.3.2.1 [IDE Compatibility Mode and Native Mode Registers]</a>
2:1	Reserved.
0	<b>ResourceTypeIndicator: Resource Type Indicator.</b> Value: 1. 1=Base address field in this register maps to the IO space.

**D11F0x1C Secondary IDE CS1 Base Address (BAR3)**

Bits	Description
31:2	<b>SecondaryIDECS1BaseAddress: Secondary IDE CS1 Base Address.</b> Read-write. Reset: 0. Base address[31:2] for Secondary IDE Bus CS1 in IDE mode. This register is used for native mode only. See <a href="#">3.25.3.2.1 [IDE Compatibility Mode and Native Mode Registers]</a>
1	Reserved.
0	<b>ResourceTypeIndicator: Resource Type Indicator.</b> Value: 1. 1=Base address field in this register maps to the IO space.

**D11F0x20 Bus Master Interface Register Base Address (BAR4)**

Bits	Description
31:4	<b>BusMasterInterfaceRegisterBaseAddress: Bus Master Interface Register Base Address.</b> Read-write. Reset: 0. Base address[31:4] for Bus Master interface registers. See <a href="#">3.25.3.2.2 [IDE Bus Master Registers]</a> .
3:1	Reserved.
0	<b>ResourceTypeIndicator: Resource Type Indicator.</b> Value: 1. 1=Base address field in this register maps to the IO space.

**D11F0x24 AHCI Base Address (BAR5)**

Bits	Description
31:10	<b>AHCIBaseAddress: AHCI Base Address[31:10]</b> . Read-write. Reset: 0. Specifies base address[31:10] of the AHCI control register space. See 3.25.3.3 [SATA Memory Mapped AHCI Registers].
9:1	Reserved.
0	<b>ResourceTypeIndicator: Resource Type Indicator</b> . Value: 0. 0=Base address field in this register maps to the memory space.

**D11F0x2C Subsystem ID and Subsystem Vendor ID**

Bits	Description
31:16	<b>SubsystemID: Subsystem ID</b> . Write-once; read. Reset: 0.
15:0	<b>SubsystemVendorID: Subsystem Vendor ID</b> . Write-once; read. Reset: 0.

**D11F0x34 Capabilities Pointer**

Bits	Description
31:8	Reserved.
7:0	<b>CapabilitiesPointer: Capabilities Pointer</b> . Read-only. Reset: 60h. The first pointer of the Capability block.

**D11F0x3C Interrupt Line**

Bits	Description
31:24	<b>MaximumLatency: Maximum Latency</b> . Value: 0. Specifies the Maximum Latency time required before the SATA controller can start an access as a bus-master.
23:16	<b>MinimumGrant: Minimum Grant</b> . Value: 0. Specifies the desired settings for how long of a burst the SATA controller needs. The value specifies a period of time in units of 1/4 microseconds.
15:8	<b>InterruptPin: Interrupt Pin</b> . Value: 1.
7:0	<b>InterruptLine: Interrupt Line</b> . Read-write. Reset: 0. Identifies which input on the interrupt controller this function's PCI interrupt request pin as specified in InterruptPin is routed to.

**D11F0x40 Misc Control**

Bits	Description
31:24	Reserved.

23:16	<b>SataPortDisable[7:0]</b> . Read-write. Reset: 0. BIOS: See 2.15.3.6.1. This field is bit-significant where bit 0 controls port 0. For each bit, 1=The corresponding port is disabled, link/transport layer clocks are shut down. <table border="1"> <thead> <tr> <th>Bit</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>[0]</td> <td>1=Port 0 disable</td> </tr> <tr> <td>[1]</td> <td>1=Port 1 disable</td> </tr> <tr> <td>[7:2]</td> <td>Reserved</td> </tr> </tbody> </table>	Bit	Definition	[0]	1=Port 0 disable	[1]	1=Port 1 disable	[7:2]	Reserved
Bit	Definition								
[0]	1=Port 0 disable								
[1]	1=Port 1 disable								
[7:2]	Reserved								
15:14	Reserved.								
13	Reserved.								
12:3	Reserved.								
2:1	Reserved.								
0	<b>SubclassCodeWriteEnable: Subclass Code Write Enable</b> . Read-write. Reset: 0. BIOS: See 2.15.3.1, 2.15.3.8.2. 1=The following registers/fields are programmable: <ul style="list-style-type: none"> <li>• D11F0x08[ProgramIF[7:0]]</li> <li>• D11F0x08[SubclassCode]</li> <li>• D11F0x50[MMC]</li> </ul>								

#### D11F0x44 Watch Dog Control And Status

Bits	Description
31:24	Reserved.
23:16	<b>WatchdogCounter: Watchdog Counter</b> . Read-write. Reset: 80h. BIOS: 20h. Specifies the timeout retry count for PCI downstream retries for SATA ports.
15:2	Reserved.
1	<b>WatchdogTimeoutStatus: Watchdog Timeout Status</b> . Read; write-1-to-clear. Reset: 0. 1=The watchdog counter has expired for PCI downstream transaction, and as a result, the transaction was aborted.
0	<b>WatchdogEnable: Watchdog Enable</b> . Read-write. Reset: 0. BIOS: 1. 1=Enable the watchdog counter for all the PCI downstream transactions for SATA ports.

#### D11F0x50 MSI Control

Bits	Description
31:24	Reserved.
23	<b>C64: MSI 64-bit Address</b> . Read-only. Reset: 1. 1=64-bit address is supported.
22:20	<b>MME: Multiple Message Enable</b> . Read-write. Reset: 0. Indicates the number of messages the HBA should assert. If the value programmed into this field exceeds the MMC field in this register, the results are undetermined.
19:17	<b>MMC: Multiple Message Capable</b> . Read-only. Reset: 2h. BIOS: See 2.15.3.8.2. Indicates the number of messages the HBA wishes to assert.
16	<b>MSIE: Message Signaled Interrupt Enable</b> . Read-write. Reset: 0. 1=MSI is enabled and the traditional pins are not used to generate interrupts. 0=MSI operation is disabled and the traditional interrupt pins are used.

15:8	<b>CapabilityNextPointer: Capability Next Pointer.</b> Read-only. Reset: D0h. BIOS: 0.
7:0	<b>CapabilityID: Capability ID.</b> Read-only. Reset: 5h. Indicates this is MSI capability ID.

**D11F0x54 MSI Address**

Bits	Description
31:2	<b>MsiAddress.</b> Read-write. Reset: 0. Specifies the lower 32 bits of the system specified message address [31:2]. Always doubleword aligned.
1:0	Reserved.

**D11F0x58 MSI Upper Address**

Bits	Description
31:0	<b>MsiUpperAddress.</b> Read-write. Reset: 0. Specifies the upper 32 bits of the system specified message address [63:32].

**D11F0x5C MSI Data**

Bits	Description
31:16	Reserved.
15:0	<b>MsiData.</b> Read-write. Reset: 0. MSI Data.

**D11F0x60 Power Management Capability**

Bits	Description												
31:27	<b>PSUP: PME Support.</b> Read-only. Reset: 8. Indicates the states the can generate PME#. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Bit</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>[0]</td> <td>1=PME# can be generated form D0. Not supported; only interrupts are used.</td> </tr> <tr> <td>[1]</td> <td>1=PME# can be generated form D1. D1 is not supported.</td> </tr> <tr> <td>[2]</td> <td>1=PME# can be generated form D2. D2 is not supported.</td> </tr> <tr> <td>[3]</td> <td>1=PME# can be generated from D3hot state.</td> </tr> <tr> <td>[4]</td> <td>1=PME# can be generated form D3cold state. D3cold is not supported.</td> </tr> </tbody> </table>	Bit	Definition	[0]	1=PME# can be generated form D0. Not supported; only interrupts are used.	[1]	1=PME# can be generated form D1. D1 is not supported.	[2]	1=PME# can be generated form D2. D2 is not supported.	[3]	1=PME# can be generated from D3hot state.	[4]	1=PME# can be generated form D3cold state. D3cold is not supported.
Bit	Definition												
[0]	1=PME# can be generated form D0. Not supported; only interrupts are used.												
[1]	1=PME# can be generated form D1. D1 is not supported.												
[2]	1=PME# can be generated form D2. D2 is not supported.												
[3]	1=PME# can be generated from D3hot state.												
[4]	1=PME# can be generated form D3cold state. D3cold is not supported.												
26	<b>D2S: D2 Support.</b> Read-only. Reset: 0. 0=D2 state is not supported.												
25	<b>D1S: D1 Support.</b> Read-only. Reset: 0. 0=D1 state is not supported.												
24:22	<b>AUXC: Aux Current.</b> Value: 0. Reports the maximum Suspend well current required in D3COLD state.												
21	<b>DSI: Device Specific Initialization.</b> Read-only. Reset: 1. 1=Device-specific initialization is required.												
20	Reserved.												
19	<b>PMEC: PME Clock.</b> Read-only. Reset: 0. 0=PCI clock is not required to generate PME#.												
18:16	<b>Version: Version.</b> Read-only. Reset: 3h. Indicates support for Revision 1.2 of the PCI Power Management Specification. .												

15:8	<b>CapabilityNextPointer: Capability Next Pointer.</b> Read-only. Reset: 70h. Points to SATA Capability.
7:0	<b>CapabilityID: Capability ID.</b> Read-only. Reset: 1. Indicates that this pointer is for PCI power management.

### D11F0x64 PCI Power Management Control And Status

Bits	Description												
31:16	Reserved.												
15	<b>PMES: PME Status.</b> Read; Set-by-hardware; Write-1-to-clear. Cold reset: 0. This bit is set independent of PMEE according to PCI IPM. Writing '0' will have no effect. 1=PME# is generated by HBA.												
14:9	Reserved.												
8	<b>PMEE: PME Enable.</b> Read-write. Reset: 0. 1=The SATA controller asserts the PME# signal when PMES==1. 0=PME# assertion is disabled.												
7:4	Reserved.												
3	<b>NSR: No soft reset.</b> Value: 0. If the device transitions from D3hot to D0 due to system or bus segment reset and PME is supported and enabled, it returns to the device D0 state uninitialized with only PME context preserved regardless of this bit. 1=Devices do not perform an internal reset upon transitioning from D3hot to D0 because of PowerState commands; Configuration contexts is still preserved; Upon transitioning from D3hot to D0 Initialized state, no additional operating system intervention is required to preserve configuration context beyond writing the PowerState bits. 0=Devices do perform an internal reset upon transitioning from D3hot to D0 via software control of the PowerState bits; Configuration context is lost when performing the soft reset; Upon transition from the D3hot to D0 state, full re-initialization sequence is needed to return the device to D0 initialized.												
2	Reserved.												
1:0	<b>PS: Power State.</b> Read-write. Reset: 0. This field is used both to determine the current power state of the HBA and to set a new power state. The D1 and D2 states are not supported. When the HBA is in D3hot state, the configuration space is available, but the register memory spaces are not; Additionally, interrupts are blocked. <table border="1" style="margin-left: 40px;"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>D0 state.</td> <td>10b</td> <td>Reserved.</td> </tr> <tr> <td>01b</td> <td>Reserved.</td> <td>11b</td> <td>D3hot state.</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	<u>Bits</u>	<u>Definition</u>	00b	D0 state.	10b	Reserved.	01b	Reserved.	11b	D3hot state.
<u>Bits</u>	<u>Definition</u>	<u>Bits</u>	<u>Definition</u>										
00b	D0 state.	10b	Reserved.										
01b	Reserved.	11b	D3hot state.										

### D11F0x70 Serial ATA Capability Register 0

Bits	Description
31:24	Reserved.
23:20	<b>MajorRevision: Major Revision.</b> Read-only. Reset: 1. Major revision number of the SATA Capability Pointer implemented.
19:16	<b>MinorRevision: Minor Revision.</b> Read-only. Reset: 0. Minor revision number of the SATA Capability Pointer implemented.

15:8	<b>CapabilityNextPointer: Capability Next Pointer.</b> Read-only. Reset: 50h. BIOS: See <a href="#">2.15.3.8.2</a> . Points to MSI capability.
7:0	<b>CapabilityID: Capability ID.</b> Read-only. Reset: 12h. Indicate that this pointer is for Serial ATA Capability.

### D11F0x74 Serial ATA Capability Register 1

Bits	Description
31:24	Reserved.
23:4	<b>BAROffset: BAR Offset.</b> Read-only. Reset: 0. Indicates the offset into the BAR where the index/data pair are located in doubleword granularity.
3:0	<b>BARLocation: BAR Location.</b> Read-only. Reset: Fh. Indicates that index/data pair is implemented in doublewords directly following SATA Capability Register 1 in the PCI configuration space.

### D11F0x78 IDP Index Register

Bits	Description
31:10	Reserved.
9:2	<b>IDPIndex: IDP Index.</b> Read-write. Reset: 0. This register selects the doubleword offset of the memory mapped AHCI register to be accessed.
1:0	Reserved.

### D11F0x7C IDP Data Register

Bits	Description
31:0	<b>IDPData: IDP Data.</b> Read-write. Reset: 0h. This register is a window through which data is read or written to the memory mapped AHCI register pointed to by the <a href="#">D11F0x78 [IDP Index Register]</a> . See <a href="#">3.25.3.3 [SATA Memory Mapped AHCI Registers]</a> . This is not a physical register, the default value for this field is 0000_0000h, once software programs a valid address in <a href="#">D11F0x78 [IDP Index Register]</a> , it switches to the register value which IDPIndex pointing to. The AHCI registers can be accessed both through <a href="#">D11F0x24 [AHCI Base Address (BAR5)]</a> and through this index/data pair.

### D11F0x7C\_x00 HBA Capabilities (CAP)

Bits	Description
31:0	Alias of <a href="#">SATAx00</a> .

### D11F0x7C\_x04 Global HBA Control (GHC)

Bits	Description
31:0	Alias of <a href="#">SATAx04</a> .



**D11F0x7C\_x08 Interrupt Status (IS)**

Bits	Description
31:0	Alias of <a href="#">SATAx08</a> .

**D11F0x7C\_x0C Ports Implemented (PI)**

Bits	Description
31:0	Alias of <a href="#">SATAx0C</a> .

**D11F0x7C\_x10 AHCI Version (VS)**

Bits	Description
31:0	Alias of <a href="#">SATAx10</a> .

**D11F0x7C\_x14 Command Completion Coalescing Control (CCC\_CTL)**

Bits	Description
31:0	Alias of <a href="#">SATAx14</a> .

**D11F0x7C\_x18 Command Completion Coalescing Ports (CCC\_PORTS)**

Bits	Description
31:0	Alias of <a href="#">SATAx18</a> .

**D11F0x7C\_x1C Enclosure Management Location (EM\_LOC)**

Bits	Description
31:0	Alias of <a href="#">SATAx1C</a> .

**D11F0x7C\_x20 Enclosure Management Control (EM\_CTL)**

Bits	Description
31:0	Alias of <a href="#">SATAx20</a> .

**D11F0x7C\_x24 HBA Capabilities Extended (CAP2)**

Bits	Description
31:0	Alias of <a href="#">SATAx24</a> .

**D11F0x7C\_x28 BIOS/OS Handoff Control and Status (BOHC)**

Bits	Description
31:0	Alias of <a href="#">SATAx28</a> .

**D11F0x7C\_x1[8,0]0 Port 1,0 Command List Base Address (PxCLB)**

Bits	Description
31:0	Alias of <a href="#">SATAx1[8,0]0</a> .

**D11F0x7C\_x1[8,0]4 Port 1,0 Command List Base Upper Address (PxCLBU)**

Bits	Description
31:0	Alias of <a href="#">SATAx1[8,0]4</a> .

**D11F0x7C\_x1[8,0]8 Port 1,0 FIS Base Address (PxFB)**

Bits	Description
31:0	Alias of <a href="#">SATAx1[8,0]8</a> .

**D11F0x7C\_x1[8,0]C Port 1,0 FIS Base Address Upper (PxFBU)**

Bits	Description
31:0	Alias of <a href="#">SATAx1[8,0]C</a> .

**D11F0x7C\_x1[9,1]0 Port 1,0 Interrupt Status (PxIS)**

Bits	Description
31:0	Alias of <a href="#">SATAx1[9,1]0</a> .

**D11F0x7C\_x1[9,1]4 Port 1,0 Interrupt Enable (PxIE)**

Bits	Description
31:0	Alias of <a href="#">SATAx1[9,1]4</a> .

**D11F0x7C\_x1[9,1]8 Port 1,0 Command and Status (PxCMD)**

Bits	Description
31:0	Alias of <a href="#">SATAx1[9,1]8</a> .

**D11F0x7C\_x1[A,2]0 Port 1,0 Task File Data (PxTFD)**

Bits	Description
31:0	Alias of SATAx1[A,2]0.

**D11F0x7C\_x1[A,2]4 Port 1,0 Signature (PxSIG)**

Bits	Description
31:0	Alias of SATAx1[A,2]4.

**D11F0x7C\_x1[A,2]8 Port 1,0 Serial ATA Status (PxSSTS)**

Bits	Description
31:0	Alias of SATAx1[A,2]8.

**D11F0x7C\_x1[A,2]C Port 1,0 Serial ATA Control (PxSCTL)**

Bits	Description
31:0	Alias of SATAx1[A,2]C.

**D11F0x7C\_x1[B,3]0 Port 1,0 Serial ATA Error (PxSERR)**

Bits	Description
31:0	Alias of SATAx1[B,3]0.

**D11F0x7C\_x1[B,3]4 Port 1,0 Serial ATA Active (PxSACT)**

Bits	Description
31:0	Alias of SATAx1[B,3]4.

**D11F0x7C\_x1[B,3]8 Port 1,0 Command Issue (PxCI)**

Bits	Description
31:0	Alias of SATAx1[B,3]8.

**D11F0x7C\_x1[B,3]C Port 1,0 SNotification (PxSNTF)**

Bits	Description
31:0	Alias of SATAx1[B,3]C.

**D11F0x7C\_x1[C,4]0 Port 1,0 FIS-based Switching Control (PxFSB)**

Bits	Description
31:0	Alias of SATAx1[C,4]0.

**D11F0x80 PHY Core Control 1**

BIOS: See 2.15.3.4.

Bits	Description												
31:9	Reserved.												
8	<b>WRALL: Write Settings To All Ports.</b> Write-only. Reset: 0. Allow software to simultaneously update all ports when writing to the port dependent PHY fine-tune setting register: <a href="#">D11F0x98</a> . 1=Writes to the PHY fine tune setting registers update all ports with the same value at the same time. 0=Writes to the PHY fine tune setting registers only update individual port selected by PN.												
7:6	Reserved.												
5:4	<b>GEN: Generation Speed.</b> Read-write. Reset: 1. Selects the Generation Speed when software reads/writes to the generation speed dependent PHY fine-tune setting register: <a href="#">D11F0x98</a> . <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Reserved</td> <td>10b</td> <td>Gen 2</td> </tr> <tr> <td>01b</td> <td>Gen 1</td> <td>11b</td> <td>Gen 3</td> </tr> </tbody> </table>	Bits	Definition	Bits	Definition	00b	Reserved	10b	Gen 2	01b	Gen 1	11b	Gen 3
Bits	Definition	Bits	Definition										
00b	Reserved	10b	Gen 2										
01b	Gen 1	11b	Gen 3										
3	Reserved.												
2:0	<b>PN: PHY Fine Tune Target Port.</b> Read-write. Reset: 0. Selects the port when software reads/writes to the port dependent PHY fine-tune setting register: <a href="#">D11F0x98</a> . <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Port 0 is selected</td> </tr> <tr> <td>01b</td> <td>Port 1 is selected</td> </tr> <tr> <td>7h-2h</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Definition	00b	Port 0 is selected	01b	Port 1 is selected	7h-2h	Reserved				
Bits	Definition												
00b	Port 0 is selected												
01b	Port 1 is selected												
7h-2h	Reserved												

**D11F0x84 PHY Core Control 2**

Bits	Description						
31	Reserved.						
30	<b>DShutDwn: PHY PLL Dynamic Shutdown.</b> Read-write. Reset: 1. Turn on/off the advanced power saving feature. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>0b</td> <td>No PLL dynamic shutdown; SATA PHY PLL consumes power even though all the ports are inactive.</td> </tr> <tr> <td>1b</td> <td>SATA PHY PLL will dynamically shutdown when all the ports are inactive. This inactivity includes Slumber Mode and port disable, but excludes Partial Mode.</td> </tr> </tbody> </table>	Bits	Definition	0b	No PLL dynamic shutdown; SATA PHY PLL consumes power even though all the ports are inactive.	1b	SATA PHY PLL will dynamically shutdown when all the ports are inactive. This inactivity includes Slumber Mode and port disable, but excludes Partial Mode.
Bits	Definition						
0b	No PLL dynamic shutdown; SATA PHY PLL consumes power even though all the ports are inactive.						
1b	SATA PHY PLL will dynamically shutdown when all the ports are inactive. This inactivity includes Slumber Mode and port disable, but excludes Partial Mode.						
29:27	Reserved.						
26	<b>S5ShadowLdDis: S5 Shadow Register Reload Disable.</b> Read-write. Reset: 1. BIOS: See 2.15.3.7. 1=Not to load S5 shadow registers back to PHY related control registers when S0 domain loses power other than cold boot. 0=Turn on shadow register reloading.						

25:3	Reserved.
2	<b>RSTB: PHY Global Reset.</b> Read-write. Reset: 1. BIOS: See 2.15.3.5. 0=Reset all port logic in PHY.
1:0	Reserved.

### D11F0x88 PHY Global Control 1

Bits	Description
31:28	<b>PIIVcoTune: PLL VCO Range Select Bits.</b> Read-write. Reset: 0. BIOS: See 2.15.3.3. PLL VCO range select bits.
27:0	Reserved.

### D11F0x8C PHY Global Control 2

Bits	Description
31:8	Reserved.
7:0	<b>PLL_CLKF: PLL Clock Divider Setting.</b> Read-write. Reset: 7Dh. BIOS: See 2.15.3.5. Feedback clock divider setting.

### D11F0x98 PHY Fine Tune PortX GenX Setting

This register is both port and generation speed sensitive. Software must program [D11F0x80\[PN\]](#) to select port and [D11F0x80\[GEN\]](#) to select generation speed before accessing this register.

Bits	Description
31:17	Reserved.
16:12	<b>TX_SLEW_CNTRL: Transmitter Output Slew Control.</b> Read-write. Reset: 3h. BIOS: See 2.15.3.4. Rise/Fall time is controlled by the combined force of TX_DRV_STR, TX_SLEW_CNTRL, and TX_DEEMPH_STR. Depending on the channel loss, the value of TX_SLEW_CNTRL and TX_DEEMPH_STR may have to be adjusted based on package and board design.
11:0	Reserved.

### D11F0x9C PortX Setting 2

Bits	Description
31:12	Reserved.

11:9	<b>RX_SQDET_TH: Squelch Detector Threshold Control.</b> Read-write. Reset: IF (D11F0x80[GEN]==1) THEN 4h. ELSEIF (D11F0x80[GEN]==2) THEN 4h. ELSEIF (D11F0x80[GEN]==3) THEN 4h. ENDIF. BIOS: See 2.15.3.4. Specifies the squelch detector threshold adjustment.			
	<u>Bits</u>	<u>Definition</u>	<u>Bits</u>	<u>Definition</u>
	000b	24 mV	100b	48 mV
	001b	30 mV	101b	54 mV
	010b	36 mV	110b	60 mV
	011b	42 mV	111b	66 mV
8:0	Reserved.			

**D11F0xB4 PortX BIST Control/Status**

Software must program D11F0xB7 [PortX BIST Port Select] to select the port before reading from or writing to this register.

Bits	Description																										
15:14	Reserved.																										
13:12	<b>PortXLinkBistSpeed: PortX Link BIST Speed.</b> Read-write. Reset: 0. PHY PortX speed control for Link BIST mode.																										
	<table border="0"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Gen 1</td> <td>10b</td> <td>Gen 2</td> </tr> <tr> <td>01b</td> <td>Gen 1</td> <td>11b</td> <td>Gen 3</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	<u>Bits</u>	<u>Definition</u>	00b	Gen 1	10b	Gen 2	01b	Gen 1	11b	Gen 3														
<u>Bits</u>	<u>Definition</u>	<u>Bits</u>	<u>Definition</u>																								
00b	Gen 1	10b	Gen 2																								
01b	Gen 1	11b	Gen 3																								
11:6	Reserved.																										
5:2	<b>PortXLinkBistPattern: PortX Link BIST Pattern.</b> Read-write. Reset: 0.																										
	<table border="0"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>0000b</td> <td>Pseudo-random with ALIGN insertion (when Error Count is used, must choose this pattern).</td> </tr> <tr> <td>0001b</td> <td>D10.2 Highest frequency (for Rx eye diagram measurement).</td> </tr> <tr> <td>0010b</td> <td>SYNC primitive (for Rx eye diagram measurement).</td> </tr> <tr> <td>0011b</td> <td>Reserved.</td> </tr> <tr> <td>0100b</td> <td>Mid Frequency Test Pattern (MFTP).</td> </tr> <tr> <td>0101b</td> <td>Reserved.</td> </tr> <tr> <td>0110b</td> <td>Reserved.</td> </tr> <tr> <td>0111b</td> <td>Forced T-mode enable. T-mode is defined as Far end transmit only mode without Device initiating. In T-mode, the BIST pattern generated is based on the values programmed in D11F0xBC and D11F0xC0.</td> </tr> <tr> <td>1000b</td> <td>Reserved.</td> </tr> <tr> <td>1001b</td> <td>Reserved.</td> </tr> <tr> <td>1010b</td> <td>Reserved.</td> </tr> <tr> <td>1111b-1011b</td> <td>Reserved.</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	0000b	Pseudo-random with ALIGN insertion (when Error Count is used, must choose this pattern).	0001b	D10.2 Highest frequency (for Rx eye diagram measurement).	0010b	SYNC primitive (for Rx eye diagram measurement).	0011b	Reserved.	0100b	Mid Frequency Test Pattern (MFTP).	0101b	Reserved.	0110b	Reserved.	0111b	Forced T-mode enable. T-mode is defined as Far end transmit only mode without Device initiating. In T-mode, the BIST pattern generated is based on the values programmed in D11F0xBC and D11F0xC0.	1000b	Reserved.	1001b	Reserved.	1010b	Reserved.	1111b-1011b	Reserved.
<u>Bits</u>	<u>Definition</u>																										
0000b	Pseudo-random with ALIGN insertion (when Error Count is used, must choose this pattern).																										
0001b	D10.2 Highest frequency (for Rx eye diagram measurement).																										
0010b	SYNC primitive (for Rx eye diagram measurement).																										
0011b	Reserved.																										
0100b	Mid Frequency Test Pattern (MFTP).																										
0101b	Reserved.																										
0110b	Reserved.																										
0111b	Forced T-mode enable. T-mode is defined as Far end transmit only mode without Device initiating. In T-mode, the BIST pattern generated is based on the values programmed in D11F0xBC and D11F0xC0.																										
1000b	Reserved.																										
1001b	Reserved.																										
1010b	Reserved.																										
1111b-1011b	Reserved.																										
1	Reserved.																										
0	<b>PortXLinkBistEnable: PortX Link BIST Enable.</b> Read-write. Reset: 0. 1=PortX is in link BIST mode overriding normal operation and device OOB is ignored by host. To resume normal operation, this bit needs to be cleared to 0 and a port reset is required.																										

**D11F0xB7 PortX BIST Port Select**

Bits	Description
7:2	Reserved.
1:0	<b>PortXBistPortSelect: PortX BIST Port Select.</b> Read-write. Reset: 0. This register is bit significant. When the bit is 1, it indicates the port that software will read from/write to during link BIST. For reads, the port is determined by the last significant bit with value 1 in this field. For writes, the port(s) is determined by the any bit with value 1 in this field. This register must be programmed before writing to/reading from <a href="#">D11F0xB4</a> to avoid reading from/written to an un-expected port.

**D11F0xBC T-Mode BIST Transit Pattern DW1**

Bits	Description
31:0	<b>TmodeBistTransitPatternDW1: T-mode BIST Transit Pattern DW1.</b> Read-write. Reset: 0. Transit Pattern DW1. See <a href="#">D11F0xB4</a> [PortXLinkBistPattern] for T-mode enable.

**D11F0xC0 T-Mode BIST Transit Pattern DW2**

Bits	Description
31:0	<b>TmodeBistTransitPatternDW2: T-mode BIST Transit Pattern DW2.</b> Read-write. Reset: 0. Transit Pattern DW2. See <a href="#">D11F0xB4</a> [PortXLinkBistPattern] for T-mode enable.

**D11F0xC4 T-Mode BIST Transit Control**

Bits	Description
31:3	Reserved.
2	<b>TmodePbit: T-mode P bit.</b> Read-write. Reset: 0. The transmit primitives bit.
1	<b>TmodeSbit: T-mode S bit.</b> Read-write. Reset: 0. Scrambling Bypass.
0	<b>TmodeAbit: T-mode A bit.</b> Read-write. Reset: 0. ALIGN primitives bypass mode.

**D11F0xD0 Advanced Features Capability Register 0**

Bits	Description
31:26	Reserved.
25	<b>FlrCap.</b> Read-only. Reset: 1. 1=Indicates support for Function Level Reset (FLR).
24	<b>TpCap.</b> Read-only. Reset: 1. 1=Indicates support for the Transactions Pending (TP) bit ( <a href="#">D11F0xD4</a> [TP]). TP must be supported if FLR is supported.
23:16	<b>Length.</b> Read-only. Reset: 6h. Advanced Feature (AF) Structure Length (Bytes).
15:8	<b>NextPtr: Next pointer.</b> Read-only. Reset: 0. 0=End of list.
7:0	<b>CapID: Capability ID.</b> Read-only. Reset: 13h. Identifies the function being Advanced Feature (AF) capable.

**D11F0xD4 Advanced Features Capability Register 1**

Bits	Description
31:16	Reserved.
15:9	Reserved.
8	<b>TP: Transactions Pending.</b> Read-only; updated-by-hardware. Reset: 0. 1=The function has issued one or more non-posted transactions which have not been completed, including non-posted transactions that a target has terminated with retry. 0=All non-posted transactions have been completed.
7:1	Reserved.
0	<b>InitiateFlr.</b> Read-write; RAZ. Reset: 0. 1=Initiate Function Level Reset (FLR).

**D11F0xE0 PCI Target Control TimeOut Counter**

Bits	Description
15:8	Reserved.
7:0	<b>Count: PCI Target Control Timeout Count.</b> Read-write. Reset: 80h. This field specifies the PCI target control timeout count used to clear any stale target commands to the host controller. Granularity is 15.5 us (Timeout Counter = Count * 15.5 us). The counter is disabled if the count is programmed to 0.

**3.25.3.2 SATA IO Registers****3.25.3.2.1 IDE Compatibility Mode and Native Mode Registers**

See 3.1 [Register Descriptions and Mnemonics] for a description of the register naming convention.

Compatibility Mode is selected by **D11F0x08[ProgramIF[0], ProgramIF[2]]==0** respectively. In Compatibility Mode the IO address is forced to the legacy compatible address. The same group of registers can be specified as Native Mode by **D11F0x08[ProgramIF[0], ProgramIF[2]]==1** respectively. In Native Mode, the IO address is specified by **D11F0x10, D11F0x14, D11F0x18** and **D11F0x1C**.

**Table 232: IDE Compatibility Mode and Native Mode Address Mapping**

Compatibility Mode: Interface, IO Address	Native Mode: Interface, BAR+Offset	Read Function	Write Function	Register
Primary, 1F0h	Primary, <b>D11F0x10+0</b>	Data (16 bit)	Data (16 bit)	IDE[0]_x00
Primary, 1F1h	Primary, <b>D11F0x10+1</b>	Error	Feature	IDE[0]_x01
Primary, 1F2h	Primary, <b>D11F0x10+2</b>	Sector Count	Sector Count	IDE[0]_x02
Primary, 1F3h	Primary, <b>D11F0x10+3</b>	Sector Number	Sector Number	IDE[0]_x03
Primary, 1F4h	Primary, <b>D11F0x10+4</b>	Cylinder Low	Cylinder Low	IDE[0]_x04
Primary, 1F5h	Primary, <b>D11F0x10+5</b>	Cylinder High	Cylinder High	IDE[0]_x05
Primary, 1F6h	Primary, <b>D11F0x10+6</b>	Drive/Head	Drive/Head	IDE[0]_x06
Primary, 1F7h	Primary, <b>D11F0x10+7</b>	Status	Command	IDE[0]_x07
Primary, 3F6h	Primary, <b>D11F0x14+2</b>	Alternate Status	Device Control	IDEA[0]_x02
Secondary, 170h	Secondary, <b>D11F0x18+0</b>	Data (16 bit)	Data (16 bit)	IDE[1]_x00



**Table 232: IDE Compatibility Mode and Native Mode Address Mapping**

Compatibility Mode: Interface, IO Address	Native Mode: Interface, BAR+Offset	Read Function	Write Function	Register
Secondary, 171h	Secondary, D11F0x18+1	Error	Feature	IDE[1]_x01
Secondary, 172h	Secondary, D11F0x18+2	Sector Count	Sector Count	IDE[1]_x02
Secondary, 173h	Secondary, D11F0x18+3	Sector Number	Sector Number	IDE[1]_x03
Secondary, 174h	Secondary, D11F0x18+4	Cylinder Low	Cylinder Low	IDE[1]_x04
Secondary, 175h	Secondary, D11F0x18+5	Cylinder High	Cylinder High	IDE[1]_x05
Secondary, 176h	Secondary, D11F0x18+6	Drive/Head	Drive/Head	IDE[1]_x06
Secondary, 177h	Secondary, D11F0x18+7	Status	Command	IDE[1]_x07
Secondary, 376h	Secondary, D11F0x1C+2	Alternate Status	Device Control	IDEA[1]_x02

**IDE[1:0]x00 IDE DATA**

Bits	Description
15:0	<b>Data.</b> Read-write. Reset: 0. Accesses to this register must be 16 bit. Addressing as an 8 bit register will result in IDE[1:0]x00[15:8] aliasing to IDE[1:0]x01.

**IDE[1:0]x01 Feature and Error**

Bits	Description
7:0	<b>FeatureError.</b> Read-write. Reset: 0.

**IDE[1:0]x02 Sector Count**

Bits	Description
7:0	<b>SectorCount.</b> Read-write. Reset: 0.

**IDE[1:0]x03 Sector Number**

Bits	Description
7:0	<b>SectorNumber.</b> Read-write. Reset: 0.

**IDE[1:0]x04 Cylinder Low**

Bits	Description
7:0	<b>CylinderLow.</b> Read-write. Reset: 0.

**IDE[1:0]x05 Cylinder High**

Bits	Description
7:0	<b>CylinderHigh</b> . Read-write. Reset: 0.

**IDE[1:0]x06 Drive and Head**

Bits	Description
7:0	<b>DriveHead</b> . Read-write. Reset: 0.

**IDE[1:0]x07 Command and Status**

Bits	Description
7:0	<b>CommandStatus</b> . Read-write. Reset: 0.

**IDEA[1:0]x02 Device Control and Alternate Status**

Bits	Description
7:0	<b>DevCtrlAltStatus</b> . Read-write. Reset: 0.

**3.25.3.2.2 IDE Bus Master Registers**

See section 3.1 [Register Descriptions and Mnemonics] for a description of the register naming convention.

**IDE\_BMx0[8,0] Bus Master IDE Command**

Primary and Secondary Bus Master IDE Command registers.

Bits	Description						
7:4	Reserved.						
3	<b>BusMasterReadWrite: Bus Master Read/Write</b> . Read-write. Reset: 0. This bit should not change during Bus Master transfer cycle, even if terminated by Bus Master IDE Stop. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>0b</td> <td>Memory to IDE.</td> </tr> <tr> <td>1b</td> <td>IDE to memory.</td> </tr> </tbody> </table>	Bits	Definition	0b	Memory to IDE.	1b	IDE to memory.
Bits	Definition						
0b	Memory to IDE.						
1b	IDE to memory.						
2:1	Reserved.						
0	<b>BusMasterIDEStartStop: Bus Master IDE Start/Stop</b> . Read-write. Reset: 0. This bit will not be reset by interrupt from IDE device. This must be reset by software (device driver). <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>0b</td> <td>Stop.</td> </tr> <tr> <td>1b</td> <td>Start.</td> </tr> </tbody> </table>	Bits	Definition	0b	Stop.	1b	Start.
Bits	Definition						
0b	Stop.						
1b	Start.						

**IDE\_BMx0[A,2] Bus-master IDE Status**

Primary and Secondary Bus Master IDE Status registers.

Bits	Description						
7	<b>SimplexOnly: Simplex Only.</b> Value: 0.						
6	<b>SlaveDeviceDMACapable: Slave Device DMA Capable.</b> Read-write. Reset: 0. Device 1 (Slave) DMA capable.						
5	<b>MasterDeviceDMACapable: Master Device DMA Capable.</b> Read-write. Reset: 0. Device 0 (Master) DMA capable.						
4:3	Reserved.						
2	<b>IDEInterrupt: IDE Interrupt.</b> Read-write. Reset: 0. 1=An IDE device has asserted its interrupt line. IRQ14 is used for the primary channel, and IRQ15 is used for the secondary channel. If the Interrupt Status bit is set to 0, by writing a 1 to this bit while the interrupt line is still at the active level, this bit will remain 0 until another assertion edge is detected on the interrupt line.						
1	<b>BusMasterDMAError: Bus Master DMA Error.</b> Read; write-1-to-clear. Reset: 0. 1=The IDE host controller encounters a target abort, master abort, or parity error while transferring data on the PCI bus.						
0	<b>BusMasterActive: Bus Master Active.</b> Read-only; updated-by-hardware. Reset: 0. Bus Master IDE active. This bit is set to 1 when <a href="#">IDE_BMx0[8,0][0]==1</a> . This bit is set to 0 when <a href="#">IDE_BMx0[8,0][0]==0</a> or the last transfer for a region is performed. <table border="0" style="margin-left: 20px;"> <tr> <td style="padding-right: 20px;"><u>Bits</u></td> <td><u>Definition</u></td> </tr> <tr> <td>0b</td> <td>Not active.</td> </tr> <tr> <td>1b</td> <td>Active.</td> </tr> </table>	<u>Bits</u>	<u>Definition</u>	0b	Not active.	1b	Active.
<u>Bits</u>	<u>Definition</u>						
0b	Not active.						
1b	Active.						

**IDE\_BMx0[C,4] Descriptor Table Pointer**

Primary and Secondary Bus Master IDE Descriptor Table Pointer register.

Bits	Description
31:2	<b>DescriptorTableBaseAddress: Descriptor Table Base Address.</b> Read-write. Reset: 0. These bits correspond to Address [31:2].
1:0	Reserved.

**3.25.3.3 SATA Memory Mapped AHCI Registers**

These are the AHCI memory mapped registers. The base address is defined by [D11F0x24 \[AHCI Base Address \(BAR5\)\]](#). See section 3.1 [\[Register Descriptions and Mnemonics\]](#) for a description of the register naming convention.

## 3.25.3.3.1 Generic Host Control Registers

**SATAx00 HBA Capabilities (CAP)**

Bits	Description												
31	<b>S64A: Supports 64-bit Addressing.</b> Read-only. Reset: 1. Indicates whether HBA can access 64-bit data structures. 1=HBA will make the 32-bit upper bits of the port DMA Descriptor, the PRD Base, and each PRD entry read/write.												
30	<b>SNCQ: Supports Native Command Queuing.</b> Read-only. Reset: 1. Indicates whether HBA supports Serial ATA native command queuing. 1=HBA can handle DMA Setup FISes natively, and can handle the auto-activate optimization through that FIS.												
29	<b>SSNTF: Supports SNotification Register.</b> Read-only. Reset: 1. 1=HBA supports the <a href="#">SATAx1[B,3]C [Port 1,0 SNotification (PxSNTF)]</a> register and its associated functionality.												
28	<b>SMPS: Supports Mechanical Presence Switch.</b> Read-only. Reset: 1. 1=HBA supports mechanical presence switches on its ports for use in hot plug operations. 0=This function is not supported. This value is loaded by the BIOS prior to OS initialization.												
27	<b>SSS: Supports Staggered Spin-up.</b> Read-only. Reset: 0. 0=This function is not supported. This value is loaded by the BIOS prior to OS initialization.												
26	<b>SALP: Supports Aggressive Link Power Management.</b> Read-only. Reset: 1. 1=HBA can support auto-generating link requests to the Partial or Slumber states when there are no commands to process.												
25	<b>SAL: Supports Activity LED.</b> Read-only. Reset: 1. 1=HBA supports a single activity indication output pin; This pin can be connected to an LED on the platform to indicate device activity on any drive.												
24	<b>SCLO: Supports Command List Override.</b> Read-only. Reset: 1. 1=HBA supports the <a href="#">SATAx1[9,1]8[CLO]</a> bit and its associated function.												
23:20	<b>ISS: Interface Speed Support.</b> Read-only. Reset: 3h. Indicates the maximum speed HBA can support on its ports. These encodings match the system software programmable <a href="#">SATAx1[A,2]C[SPD]</a> field. Values are: <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>0000b</td> <td>Reserved</td> </tr> <tr> <td>0001b</td> <td>Gen1 (1.5 Gbps)</td> </tr> <tr> <td>0010b</td> <td>Gen1 and Gen2 (3 Gbps)</td> </tr> <tr> <td>0011b</td> <td>Gen1 and Gen2 and Gen3 (6 Gbps)</td> </tr> <tr> <td>1111b-0100b</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Definition	0000b	Reserved	0001b	Gen1 (1.5 Gbps)	0010b	Gen1 and Gen2 (3 Gbps)	0011b	Gen1 and Gen2 and Gen3 (6 Gbps)	1111b-0100b	Reserved
Bits	Definition												
0000b	Reserved												
0001b	Gen1 (1.5 Gbps)												
0010b	Gen1 and Gen2 (3 Gbps)												
0011b	Gen1 and Gen2 and Gen3 (6 Gbps)												
1111b-0100b	Reserved												
19	<b>SNZO: Supports Non-Zero DMA Offsets.</b> Read-only. Reset: 0. 0=Indicates HBA can not support non-zero DMA offsets for DMA Setup FISes.												
18	<b>SAM: Supports AHCI mode only.</b> Read-only. Reset: 0. The SATA controller may optionally support AHCI access mechanisms only. 0=Indicates that in addition to the native AHCI mechanism (via AHCI BAR5), the SATA controller implements a legacy, task-file based register interface such as SFF-8038i. 1=Indicates the SATA controller does not implement a legacy, task-file based register interface.												
17	<b>SPM: Supports Port Multiplier.</b> Read-only. Reset: 1. Indicates whether HBA can support a port multiplier. 1=A port multiplier using command-based switching is supported. 0=A port multiplier is not supported, and a port multiplier may not be attached to this HBA.												
16	<b>FBSS: FIS-based Switching Supported.</b> Read-only. Reset: 1. 1=Indicates HBA supports Port Multiplier FIS-based switching. 0=Indicates HBA does not support FIS-based switching.												

15	<b>PMD: PIO Multiple DRQ Block.</b> Read-only. Reset: 1. 0=HBA only supports single DRQ block data transfers for the PIO command protocol. 1=HBA supports multiple DRQ block data transfers for the PIO command protocol.
14	<b>SSC: Slumber State Capable.</b> Read-only. Reset: 1. Indicates whether HBA can support transitions to the Slumber state. 0=Software must not allow the HBA to initiate transitions to the Slumber state via aggressive link power management nor the <a href="#">SATAx1[9,1]8[ICC]</a> field in each port, and the <a href="#">SATAx1[A,2]C[IPM]</a> field in each port must be programmed to disallow device initiated Slumber requests. 1=HBA and device initiated Slumber requests can be supported.
13	<b>PSC: Partial State Capable.</b> Read-only. Reset: 1. Indicates whether HBA can support transitions to the Partial state. 0=Software must not allow HBA to initiate transitions to the Partial state via aggressive link power management nor the <a href="#">SATAx1[9,1]8[ICC]</a> field in each port, and the <a href="#">SATAx1[A,2]C[IPM]</a> field in each port must be programmed to disallow device initiated Partial requests. 1=HBA and device initiated Partial requests can be supported.
12:8	<b>NCS: Number of Command Slots.</b> Read-only. Reset: 1Fh. 0's based value indicate the number of command slots per port supported by this HBA. A minimum of 1 and a maximum of 32 slots per port can be supported. The same number of command slots is available on each implemented port.
7	<b>CCCS: Command Completion Coalescing Supported.</b> Read-only. Reset: 1. 0=Indicates that HBA does not support command completion coalescing and <a href="#">SATAx14 [Command Completion Coalescing Control (CCC_CTL)]</a> and <a href="#">SATAx18 [Command Completion Coalescing Ports (CCC_PORTS)]</a> are not implemented. 1=Indicates that HBA supports command completion coalescing and HBA has implemented the <a href="#">SATAx14 [Command Completion Coalescing Control (CCC_CTL)]</a> and the <a href="#">SATAx18 [Command Completion Coalescing Ports (CCC_PORTS)]</a> global HBA registers.
6	<b>EMS: Enclosure Management Supported.</b> Read-only. Reset: 0b. 1=Indicates that HBA supports enclosure management and has implemented the <a href="#">SATAx1C [Enclosure Management Location (EM_LOC)]</a> and <a href="#">SATAx20 [Enclosure Management Control (EM_CTL)]</a> global HBA registers. 0=Indicates that the HBA does not support enclosure management and the <a href="#">SATAx1C [Enclosure Management Location (EM_LOC)]</a> and <a href="#">SATAx20 [Enclosure Management Control (EM_CTL)]</a> global HBA registers are not implemented.
5	<b>SXS: Supports External SATA.</b> Read-only. Reset: 0. 1=Indicates that HBA has one or more Serial ATA ports that have a signal-only connector (i.e. power is not part of that connector) that is externally accessible. 0=Indicates that the HBA has no Serial ATA ports that have a signal-only connector externally accessible. If this bit is set to 1, software may refer to the <a href="#">SATAx1[9,1]8[ESP]</a> bit to determine whether a specific port has its signal connector externally accessible as a signal-only connector.
4:0	<b>NP: Number of Ports.</b> Read-only. Reset: 3h. 0's based value indicating the maximum number of ports supported by the HBA silicon. A maximum of 32 ports can be supported. A value of 0, indicating one port, is the minimum requirement. Note that the number of ports indicated in this field may be more than the number of ports indicated in the <a href="#">SATAx0C[PI]</a> .

**SATAx04 Global HBA Control (GHC)**

Bits	Description
31	<b>AE: AHCI Enable.</b> Read-write. Reset: 0. 1=Indicates that communication to HBA shall be via AHCI mechanisms. 0=Software shall only communicate with HBA using legacy mechanisms; FISes are not posted to memory, and no commands are sent via AHCI mechanisms. This can be used by an HBA that supports both legacy mechanisms and AHCI to know when the HBA is running under an AHCI driver. Software shall set this bit to 1 before accessing other AHCI registers.
30:3	Reserved.

2	<p><b>MRSM: MSI Revert to Single Message.</b> Read-only. Reset: 0. 1=HBA requested more than one MSI vector but has reverted to using the first vector only. 0=HBA has not reverted to single MSI mode (i.e., hardware is already in single MSI mode, software has allocated the number of messages requested, or hardware is sharing interrupt vectors if <code>D11F0x50[MME]&lt;D11F0x50[MMC]</code>). HBA may revert to single MSI mode when the number of vectors allocated by the host is less than the number requested. This bit shall only be set to 1 when the following conditions hold:</p> <ul style="list-style-type: none"> <li>• <code>D11F0x50[MSIE]=1</code> (MSI is enabled)</li> <li>• <code>D11F0x50MMC]&gt;0</code> (multiple messages requested)</li> <li>• <code>D11F0x50MME]&gt;0</code> (more than one message allocated)</li> <li>• <code>D11F0x50MME]!&lt;D11F0x50[MMC]</code> (messages allocated not equal to number requested)</li> </ul> <p>When this bit is set to 1, single MSI mode operation is in use and software is responsible for clearing bits in the IS register to clear interrupts. This bit shall be cleared to 0 by hardware when any of the four conditions stated is false. This bit is also cleared to 0 when <code>D11F0x50[MSIE] = 1</code> and <code>D11F0x50[MME] = 0h</code>. In this case, hardware has been programmed to use single MSI mode, and is not “reverting” to that mode.</p>
1	<p><b>IE: Interrupt Enable.</b> Read-write. Reset: 0. This global bit enables interrupts from HBA. 0=All interrupt sources from all ports are disabled. 1=Interrupts are enabled.</p>
0	<p><b>HR: HBA Reset.</b> Read; write-1-only; cleared-by-hardware. Reset: 0. When set by software, this bit causes an internal reset of HBA. All state machines that relate to data transfers and queuing shall return to an idle condition, and all ports shall be re-initialized via COMRESET (if staggered spin-up is not supported). If staggered spin-up is supported, then it is the responsibility of software to spin-up each port after the reset has completed. When HBA has performed the reset action, it shall reset this bit to 0.</p>

**SATAx08 Interrupt Status (IS)**

Bits	Description										
31:0	<p><b>IPS: Interrupt Pending Status.</b> Read; write-1-to-clear. Reset: 0. This register is bit significant where bit 0 corresponds to port 0. For each bit, 1=The corresponding port has an interrupt pending. Software can use this information to determine which ports require service after an interrupt. The <code>IPS[X]</code> bit is only defined for ports that are implemented or for the command completion coalescing interrupt defined by <code>SATAx14[INT]</code>. All other bits are reserved.</p> <table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; border-bottom: 1px solid black;">Bit</th> <th style="text-align: left; border-bottom: 1px solid black;">Definition</th> </tr> </thead> <tbody> <tr> <td>[0]</td> <td>Port 0 has pending interrupt</td> </tr> <tr> <td>[1]</td> <td>Port 1 has pending interrupt</td> </tr> <tr> <td>...</td> <td>...</td> </tr> <tr> <td>[31]</td> <td>Port 31 has pending interrupt</td> </tr> </tbody> </table>	Bit	Definition	[0]	Port 0 has pending interrupt	[1]	Port 1 has pending interrupt	...	...	[31]	Port 31 has pending interrupt
Bit	Definition										
[0]	Port 0 has pending interrupt										
[1]	Port 1 has pending interrupt										
...	...										
[31]	Port 31 has pending interrupt										

**SATAx0C Ports Implemented (PI)**

Bits	Description								
31:0	<p><b>PI[31:0]: Port Implemented.</b> Read-only. Reset: 3h. This register is bit significant. For each bit, 1=The corresponding port is available for software to use. 0=The port is not available for software to use. The maximum number of bits set to 1 shall not exceed <code>SATAx00[NP] + 1</code>, although the number of bits set in this register may be fewer than <code>SATAx00[NP] + 1</code>. At least one bit should be set to 1. This field is loaded by BIOS.</p> <table border="1"> <thead> <tr> <th>Bit</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>[0]</td> <td>Port 0 Implemented</td> </tr> <tr> <td>[1]</td> <td>Port 1 Implemented</td> </tr> <tr> <td>[31:2]</td> <td>Reserved</td> </tr> </tbody> </table>	Bit	Definition	[0]	Port 0 Implemented	[1]	Port 1 Implemented	[31:2]	Reserved
Bit	Definition								
[0]	Port 0 Implemented								
[1]	Port 1 Implemented								
[31:2]	Reserved								

**SATAx10 AHCI Version (VS)**

Bits	Description
31:16	<b>MJR: Major Version Number.</b> Read-only. Reset: 1. Indicates the major version is 1.
15:0	<b>MNR: Minor Version Number.</b> Read-only. Reset: 300h. Indicates the minor version is 10.

**SATAx14 Command Completion Coalescing Control (CCC\_CTL)**

Bits	Description
31:16	<p><b>TV: Timeout Value.</b> Read-write. Reset: 1. The timeout value is specified in 1 millisecond intervals. The timer accuracy shall be within 5%. <code>hCccTimer</code> is loaded with this timeout value. <code>hCccTimer</code> is only decremented when commands are outstanding on selected ports. The HBA will signal a CCC interrupt when <code>hCccTimer</code> has decremented to 0. <code>hCccTimer</code> is reset to the timeout value on the assertion of each CCC interrupt. A timeout value of 0 is reserved.</p>
15:8	<p><b>CC: Command Completions.</b> Read-write. Reset: 1. Specifies the number of command completions that are necessary to cause a CCC interrupt. The HBA has an internal command completion counter, <code>hCccComplete</code> that is incremented by one each time a selected port has a command completion. When <code>hCccComplete</code> is equal to the command completions value, a CCC interrupt is signaled. The internal command completion counter is reset to 0 on the assertion of each CCC interrupt. 0=Disable CCC interrupts being generated based on the number of commands completed, i.e. CCC interrupts are only generated based on the timer in this case.</p>
7:3	<p><b>INT: CCC Interrupt.</b> Read-only. Reset: 2h. Specifies the interrupt used by the Command Completion Coalescing (CCC) feature. This interrupt must be marked as unused in <a href="#">SATAx0C [Ports Implemented (PI)]</a> by having the corresponding bit being set to 0. Thus, the CCC interrupt corresponds to the interrupt for an un-implemented port on the controller. When a CCC interrupt occurs, the corresponding bit in <a href="#">SATAx08 [Interrupt Status (IS)]</a> shall be asserted to 1. This field also specifies the interrupt vector used for MSI.</p>

2:1	Reserved.
0	<b>EN: CCC CTL Enable.</b> Read-write. Reset: 0. 0=The Command Completion Coalescing (CCC) feature is disabled and no CCC interrupts are generated. 1=The CCC feature is enabled and CCC interrupts may be generated based on timeout or command completion conditions. Software shall only change the contents of the TV and CC fields in this register when this bit is cleared to 0. On transition of this bit from 0 to 1, any updated values for the TV and CC fields shall take effect.

### SATAx18 Command Completion Coalescing Ports (CCC\_PORTS)

Bits	Description
31:0	<b>PRT: Ports.</b> Read-write. Reset: 0. This register is bit significant. Each bit corresponds to a particular port, where bit 0 corresponds to port 0. If a bit is set to 1, the corresponding port is part of the command completion coalescing feature. If a bit is cleared to 0, the port is not part of the command completion coalescing feature. Bits set to 1 in this register must also have the corresponding bit set to 1 in the <a href="#">SATAx0C [Ports Implemented (PI)]</a> . An updated value for this field shall take effect within one timer increment (1 millisecond).

### SATAx1C Enclosure Management Location (EM\_LOC)

Bits	Description
31:16	<b>OFST: Offset.</b> Value: 0000h. Specifies the offset of the message buffer in doublewords from the beginning of BAR5 defined by <a href="#">D11F0x24</a> .
15:0	<b>SZ: Buffer Size.</b> Value: 0000h. Specifies the size of the transmit message buffer area in doublewords. If both transmit and receive buffers are supported, then the transmit buffer begins at <a href="#">SATAx1C[OFST]*4</a> and the receive buffer directly follows it. If both transmit and receive buffers are supported, both buffers are of the size indicated in the Buffer Size field.

### SATAx20 Enclosure Management Control (EM\_CTL)

Bits	Description
31:28	Reserved.
27	<b>ATTR_PM: Port Multiplier Support.</b> Read-only. Reset: 0. 1=The HBA supports enclosure management messages for devices attached via a Port Multiplier. 0=The HBA does not. When cleared to 0, software should use the Serial ATA enclosure management bridge that is built into many Port Multipliers for enclosure services with these devices.
26	<b>ATTR_ALHD: Activity LED Hardware Driven.</b> Read-only. Reset: 0. 1=The HBA drives the activity LED for the LED message type in hardware and does not utilize software settings for this LED. The HBA does not begin transmitting the hardware based activity signal until after software has written CTL_TM to 1 after a reset condition.
25	<b>ATTR_XMT: Transmit Only.</b> Value: 0b. 1=The HBA only supports transmit messages and does not support receive messages. 0=The HBA supports transmit and receive messages.
24	<b>ATTR_SMB: Single Message Buffer.</b> Value: 0b. 1=The HBA has one message buffer that is shared for messages to transmit and messages received; Unsolicited receive messages are not supported and it is software's responsibility to manage access to this buffer. 0=There are separate receive and transmit buffers such that unsolicited messages could be supported.



23:20	Reserved.
19	<b>SUPP_SGPIO: SGPIO Enclosure Management Messages.</b> Value: 0b. 1=The HBA supports the SGPIO register interface message type.
18	<b>SUPP_SES2: SES-2 Enclosure Management Messages.</b> Read-only. Reset: 0. 1=The HBA supports the SES-2 message type.
17	<b>SUPP_SAFTE: SAF-TE Enclosure Management Messages.</b> Read-only. Reset: 0. 1=The HBA supports the SAF-TE message type.
16	<b>SUPP_LED: LED Message Types.</b> Read-only. Reset: 0. 1=The HBA supports the LED message type.
15:10	Reserved.
9	<b>CTL_RST: Reset.</b> Read-only. Reset: 0.
8	<b>CTL_TM: Transmit Message.</b> Read-only. Reset: 0.
7:1	Reserved.
0	<b>STS_MR: Message Received.</b> Read; write-1-to-clear. Reset: 0. The HBA sets this bit to a 1 when a message is completely received into the message buffer.

### SATAx24 HBA Capabilities Extended (CAP2)

Bits	Description
31:6	Reserved.
5	<b>DESO: DevSleep Entrance from Slumber Only.</b> Read-only. Reset: 0. 1=The HBA shall ignore software directed entrance to DevSleep via <a href="#">SATAx1[9,1]8[ICC]</a> unless [ <a href="#">SATAx1[A,2]8IPM</a> ] = 6h. 0=The HBA may enter DevSleep from any link state (active, Partial, or Slumber).
4	<b>SADM: Supports Aggressive Device Sleep Management.</b> Read-only. Reset: 0. 1=The HBA supports hardware assertion of the DEVSLP signal after the idle timeout expires. 0=This function is not supported and software shall treat the PxDEVSLP.ADSE field as reserved.
3	<b>SDS: Supports Device Sleep.</b> Read-only. Reset: 0. 1=The HBA supports the Device Sleep feature. 0=DEVSLP is not supported and software shall not set <a href="#">SATAx1[9,1]8[ICC]</a> to 8.
2:1	Reserved
0	<b>BOH: BIOS/OS Handoff.</b> Read-only. Reset: 0. 1=The HBA supports the BIOS/OS handoff mechanism and the HBA has implemented <a href="#">SATAx28 [BIOS/OS Handoff Control and Status (BOHC)]</a> . 0=The HBA does not support the BIOS/OS handoff mechanism and <a href="#">SATAx28 [BIOS/OS Handoff Control and Status (BOHC)]</a> is not implemented.

### SATAx28 BIOS/OS Handoff Control and Status (BOHC)

Bits	Description
31:5	Reserved.
4	<b>BB: BIOS Busy.</b> Read-write. Reset: 0. This bit is used by the BIOS to indicate that it is busy cleaning up for ownership change.
3	<b>OOC: OS Ownership Change.</b> Read; write-1-to-clear. Reset: 0. This bit is set to 1 when the OOS bit transitions from 0 to 1.

2	<b>SOOE: SMI on OS Ownership Change Enable.</b> Read-write. Reset: 0. 1=Enables an SMI when the OOC bit has been set to 1.
1	<b>OOS: OS Owned Semaphore.</b> Read-write. Reset: 0. The system software sets this bit to request ownership of the HBA controller. Ownership is obtained when this bit reads 1 and the BOS bit reads 0.
0	<b>BOS: BIOS Owned Semaphore.</b> Read-write. Reset: 0. The BIOS sets this bit to establish ownership of the HBA controller. BIOS will clear this bit in response to a request for ownership of the HBA by system software via OOS.

### 3.25.3.3.2 Port Control Registers

The following registers configure the SATA ports and are implemented per port. Each port shall have the same register mapping. Port 0 starts at 100h. Port 1 starts at 180h. An algorithm to determine the offset is as follows: Port offset = 100h + (Asserted Bit Position in **SATAx0C [Ports Implemented (PI)]** \* 80h).

#### SATAx1[8,0]0 Port 1,0 Command List Base Address (PxCLB)

Bits	Description
31:10	<b>CLB[31:10]: Command List Base Address [31:10].</b> Read-write. Reset: 0. Specifies the lower 32 bits of the physical base address for the command list for this port. This base address is used by HBA to fetch commands to execute. The structure pointed to by this address range is 1K-bytes in length. This address must be 1K-byte aligned.
9:0	Reserved.

#### SATAx1[8,0]4 Port 1,0 Command List Base Upper Address (PxCLBU)

Bits	Description
31:0	<b>CLBU: Command List Base Address Upper.</b> Read-write. Reset: 0. Specifies the upper 32 bits of the physical base address for the command list for this port. If 64-bit addressing is not supported, this register reads 0.

#### SATAx1[8,0]8 Port 1,0 FIS Base Address (PxFB)

Bits	Description
31:8	<b>FB[31:8]: FIS Base Address [31:8].</b> Read-write. Reset: 0. Specifies the lower 32 bits of the physical base address for the received FISes for this port. The structure pointed to by this address range is 256 bytes in length. This address must be 256-byte aligned.
7:0	Reserved.

#### SATAx1[8,0]C Port 1,0 FIS Base Address Upper (PxFBU)

Bits	Description
31:0	<b>FBU: FIS Base Address Upper.</b> Read-write. Reset: 0. Specifies the upper 32 bits of the physical base address for the received FISes for this port. If 64-bit addressing is not supported, this register reads 0.

**SATAx1[9,1]0 Port 1,0 Interrupt Status (PxIS)**

Bits	Description
31	<b>CPDS: Cold Port Detect Status.</b> Read; write-1-to-clear. Reset: 0.
30	<b>TFES: Task File Error Status.</b> Read; write-1-to-clear. Reset: 0. 1=The error bit at bit [0] of the status field in the received FIS is set to 1 when the status register is updated by the device.
29	<b>HBFS: Host Bus Fatal Error Status.</b> Read; write-1-to-clear. Reset: 0. 1=The HBA has encountered a host bus error that it cannot recover from, such as a bad software pointer. In PCI, it indicates a target or master abort.
28	<b>HBDS: Host Bus Data Error Status.</b> Read; write-1-to-clear. Reset: 0. 1=The HBA has encountered a data error (uncorrectable ECC / parity) when reading from or writing to system memory.
27	<b>IFS: Interface Fatal Error Status.</b> Read; write-1-to-clear. Reset: 0. 1=The HBA has encountered an error on the Serial ATA interface, which caused the transfer to stop.
26	<b>INFS: Interface Non-fatal Error Status.</b> Read; write-1-to-clear. Reset: 0. 1=The HBA has encountered an error on the Serial ATA interface and was able to continue operation.
25	Reserved.
24	<b>OFS: Overflow Status.</b> Read; write-1-to-clear. Reset: 0. 1=The HBA has received more bytes from a device than what was specified in the PRD table for the command.
23	<b>IPMS: Incorrect Port Multiplier Status.</b> Read; write-1-to-clear. Reset: 0. 1=The HBA has received a FIS from a device whose Port Multiplier field did not match what was expected. The IPMS bit may be set during enumeration of devices on a Port Multiplier due to the normal Port Multiplier enumeration process. It is recommended that IPMS only be used after enumeration is complete on the Port Multiplier.
22	<b>PRCS: PhyRdy Change Status.</b> Read-only. Reset: 0. 1=The internal PhyRdy signal has changed state. This bit reflects the state of <a href="#">SATAx1[B,3]0[DIAG]</a> . To clear this bit, software must program <a href="#">SATAx1[B,3]0[DIAG]</a> to 0.
21:8	Reserved.
7	<b>DMPS: Device Mechanical Presence Status.</b> Read; write-1-to-clear. Reset: 0. 1=A mechanical presence switch attached to this port has been opened or closed, which may lead to a change in the connection state of the device. This bit is only valid if (( <a href="#">SATAx00[SMPS]</a> ==1) && ( <a href="#">SATAx1[9,1]8[MPSP]</a> ==1)).
6	<b>PCS: Port Connect Change Status.</b> Read-only. Reset: 0. 0=No change in Current Connect Status. 1=Change in Current Connect Status. This bit reflects the state of <a href="#">SATAx1[B,3]0[DIAG[10]]</a> . This bit is only cleared when <a href="#">SATAx1[B,3]0[DIAG]</a> is cleared.
5	<b>DPS: Descriptor Processed.</b> Read; write-1-to-clear. Reset: 0. 1=A Physical Region Descriptor (PRD) with the I bit set has transferred all of its data.
4	<b>UFS: Unknown FIS Interrupt.</b> Read-only. Reset: 0. 1=An unknown FIS was received with the I bit set and has been copied into system memory. This bit is cleared to 0 by software clearing the <a href="#">SATAx1[B,3]0[DIAG]</a> bit to 0. This bit does not directly reflect the <a href="#">SATAx1[B,3]0[DIAG]</a> bit. <a href="#">SATAx1[B,3]0[DIAG]</a> is set immediately when an unknown FIS is detected, whereas this bit is set when that FIS is posted to memory. Software should wait to act on an unknown FIS until this bit is set to 1, or the two bits may become out-of- sync.
3	<b>SDBS: Set Device Bits Interrupt.</b> Read; write-1-to-clear. Reset: 0. 1=A Set Device Bits FIS has been received with the I bit set and has been copied into system memory.

2	<b>DSS: DMA Setup FIS Interrupt.</b> Read-write. Reset: 0. 1=A DMA Setup FIS has been received with the I bit set and has been copied into system memory.
1	<b>PSS: PIO Setup FIS Interrupt.</b> Read; write-1-to-clear. Reset: 0. 1=A PIO Setup FIS has been received with the I bit set, and it has been copied into system memory, and the data related to that FIS has been transferred. This bit shall be set even if the data transfer resulted in an error.
0	<b>DHRS: Device to Host Register FIS Interrupt.</b> Read; write-1-to-clear. Reset: 0. 1=A D2H Register FIS has been received with the I bit set, and has been copied into system memory.

### SATAx1[9,1]4 Port 1,0 Interrupt Enable (PxIE)

Bits	Description
31	<b>CPDE: Cold Presence Detect Enable.</b> Read-only. Reset: 0.
30	<b>TFEE: Task File Error Enable.</b> Read-write. Reset: 0. 1=The HBA generates an interrupt if ((SATAx04[IE]==1) && (SATAx1[9,1]0[TFES]==1)).
29	<b>HBFE: Host Bus Fatal Error Enable.</b> Read-write. Reset: 0. 1=The HBA generates an interrupt if ((SATAx04[IE]==1) && (SATAx1[9,1]0[HBFS]==1)).
28	<b>HBDE: Host Bus Data Error Enable.</b> Read-write. Reset: 0. 1=The HBA generates an interrupt if ((SATAx04[IE]==1) && SATAx1[9,1]0[HBDS]==1)).
27	<b>IFE: Interface Fatal Error Enable.</b> Read-write. Reset: 0. 1=The HBA generates an interrupt if ((SATAx04[IE]==1) && (SATAx1[9,1]0[IFS]==1)).
26	<b>INFE: Interface Non-fatal Error Enable.</b> Read-write. Reset: 0. 1=The HBA generates an interrupt if ((SATAx04[IE]==1) && (SATAx1[9,1]0[INFS]==1)).
25	Reserved.
24	<b>OFE: Overflow Enable.</b> Read-write. Reset: 0. 1=The HBA generates an interrupt if ((SATAx04[IE]==1) && (SATAx1[9,1]0[OFS]==1)).
23	<b>IPME: Incorrect Port Multiplier Enable.</b> Read-write. Reset: 0. 1=The HBA generates an interrupt if ((SATAx04[IE]==1) && (SATAx1[9,1]0[IPMS]==1)).
22	<b>PRCE: PhyRdy Change Interrupt Enable.</b> Read-write. Reset: 0. 1=The HBA generates an interrupt if ((SATAx04[IE]==1) && (SATAx1[9,1]0[PRCS]==1)).
21:8	Reserved.
7	<b>DMPE: Device Mechanical Presence Enable.</b> Read-write. Reset: 0. 1=The HBA generates an interrupt if ((SATAx04[IE]==1) && (SATAx1[9,1]0[DMPS]==1)). For systems that do not support a mechanical presence switch, this bit will be read-only and will return a 0.
6	<b>PCE: Port Change Interrupt Enable.</b> Read-write. Reset: 0. 1=The HBA generates an interrupt if ((SATAx04[IE]==1) && (SATAx1[9,1]0[PCS]==1)).
5	<b>DPE: Descriptor Processed Interrupt Enable.</b> Read-write. Reset: 0. 1=The HBA generates an interrupt if ((SATAx04[IE]==1) && (SATAx1[9,1]0[DPS]==1)).
4	<b>UFE: Unknown FIS Interrupt Enable.</b> Read-write. Reset: 0. 1=The HBA generates an interrupt if ((SATAx04[IE]==1) && (SATAx1[9,1]0[UFS]==1)).
3	<b>SDBE: Set Device Bits FIS Interrupt Enable.</b> Read-write. Reset: 0. 1=The HBA generates an interrupt if ((SATAx04[IE]==1) && (SATAx1[9,1]0[SDBS]==1)).
2	<b>DSE: DMA Setup FIS Interrupt Enable.</b> Read-write. Reset: 0. 1=The HBA generates an interrupt if ((SATAx04[IE]==1) && (SATAx1[9,1]0[DSS]==1)).

1	<b>PSE: PIO Setup FIS Interrupt Enable.</b> Read-write. Reset: 0. 1=The HBA generates an interrupt if ((SATAx04[IE]==1) && (SATAx1[9,1]0[PSS]==1)).
0	<b>DHRE: Device to Host Register FIS Interrupt Enable.</b> Read-write. Reset: 0. 1=The HBA generates an interrupt if ((SATAx04[IE]==1) && (SATAx1[9,1]0[DHRS]==1)).

**SATAx1[9,1]8 Port 1,0 Command and Status (PxCMD)**

Bits	Description																		
31:28	<p><b>ICC: Interface Communication Control.</b> Read-write. Reset: 0. This field is used to control power management states of the interface. If the Link layer is currently in the L_IDLE state, writing to this field shall cause the HBA to initiate a transition to the interface power management state requested. If the Link layer is not currently in the L_IDLE state, writing to this field shall have no effect.</p> <table border="0" style="width: 100%;"> <thead> <tr> <th style="text-align: left;"><u>Bits</u></th> <th style="text-align: left;"><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>Fh-9h</td> <td>Reserved</td> </tr> <tr> <td>8h</td> <td>DevSleep. This shall cause the HBA to assert the DEVSLP signal associated with the port; the HBA shall ignore the Device Sleep Idle Timeout value specified by PxDEVSLP.DITO. Software shall only request DevSleep when the interface is in an idle state (i.e. PxCI is cleared to 0h and PxSACT are cleared to 0h); if CAP2.SDS is cleared to '0' or if the interface is not idle at the time the register is written, then the HBA shall not assert the DEVSLP signal and the interface remains in its current state. If CAPS.SDS is set to '1', CAP2.DESO is set to '1', and PxSSTS.IPM is not set to '6h', then the HBA shall not assert the DEVSLP signal and the interface shall remain in its current state. Additionally, the HBA shall not assert the DEVSLP signal until PHYRDY has been achieved (after a previous de-assertion).</td> </tr> <tr> <td>7h</td> <td>Reserved</td> </tr> <tr> <td>6h</td> <td>Slumber: This shall cause the HBA to request a transition of the interface to the Slumber state. The SATA device may reject the request and the interface shall remain in its current state.</td> </tr> <tr> <td>5h-3h</td> <td>Reserved</td> </tr> <tr> <td>2h</td> <td>Partial: This shall cause the HBA to request a transition of the interface to the Partial state. The SATA device may reject the request and the interface shall remain in its current state.</td> </tr> <tr> <td>1h</td> <td>Active: This shall cause the HBA to request a transition of the interface into the active state.</td> </tr> <tr> <td>0h</td> <td>No-Op, Idle: When software reads this value, it indicates the HBA is ready to accept a new interface control command, although the transition to the previously selected state may not have occurred yet.</td> </tr> </tbody> </table> <p>When system software writes a non-reserved value other than No-Op (0h), the HBA shall perform the action and update this field back to Idle (0h). If software writes to this field to change the state to a state the link is already in (i.e. interface is in the active state and a request is made to go to the active state), the HBA shall take no action and return this field to Idle. If the interface is in a low power state and software wants to transition to a different low power state, software must first bring the link to active and then initiate the transition to the desired low power state with the exception of DEVSLP. The transition to DevSleep may occur from any other state if SATAx24[DESO]==0. If SATAx24[DESO]==1, then DevSleep may only be transitioned to if the link is in Slumber.</p>	<u>Bits</u>	<u>Definition</u>	Fh-9h	Reserved	8h	DevSleep. This shall cause the HBA to assert the DEVSLP signal associated with the port; the HBA shall ignore the Device Sleep Idle Timeout value specified by PxDEVSLP.DITO. Software shall only request DevSleep when the interface is in an idle state (i.e. PxCI is cleared to 0h and PxSACT are cleared to 0h); if CAP2.SDS is cleared to '0' or if the interface is not idle at the time the register is written, then the HBA shall not assert the DEVSLP signal and the interface remains in its current state. If CAPS.SDS is set to '1', CAP2.DESO is set to '1', and PxSSTS.IPM is not set to '6h', then the HBA shall not assert the DEVSLP signal and the interface shall remain in its current state. Additionally, the HBA shall not assert the DEVSLP signal until PHYRDY has been achieved (after a previous de-assertion).	7h	Reserved	6h	Slumber: This shall cause the HBA to request a transition of the interface to the Slumber state. The SATA device may reject the request and the interface shall remain in its current state.	5h-3h	Reserved	2h	Partial: This shall cause the HBA to request a transition of the interface to the Partial state. The SATA device may reject the request and the interface shall remain in its current state.	1h	Active: This shall cause the HBA to request a transition of the interface into the active state.	0h	No-Op, Idle: When software reads this value, it indicates the HBA is ready to accept a new interface control command, although the transition to the previously selected state may not have occurred yet.
<u>Bits</u>	<u>Definition</u>																		
Fh-9h	Reserved																		
8h	DevSleep. This shall cause the HBA to assert the DEVSLP signal associated with the port; the HBA shall ignore the Device Sleep Idle Timeout value specified by PxDEVSLP.DITO. Software shall only request DevSleep when the interface is in an idle state (i.e. PxCI is cleared to 0h and PxSACT are cleared to 0h); if CAP2.SDS is cleared to '0' or if the interface is not idle at the time the register is written, then the HBA shall not assert the DEVSLP signal and the interface remains in its current state. If CAPS.SDS is set to '1', CAP2.DESO is set to '1', and PxSSTS.IPM is not set to '6h', then the HBA shall not assert the DEVSLP signal and the interface shall remain in its current state. Additionally, the HBA shall not assert the DEVSLP signal until PHYRDY has been achieved (after a previous de-assertion).																		
7h	Reserved																		
6h	Slumber: This shall cause the HBA to request a transition of the interface to the Slumber state. The SATA device may reject the request and the interface shall remain in its current state.																		
5h-3h	Reserved																		
2h	Partial: This shall cause the HBA to request a transition of the interface to the Partial state. The SATA device may reject the request and the interface shall remain in its current state.																		
1h	Active: This shall cause the HBA to request a transition of the interface into the active state.																		
0h	No-Op, Idle: When software reads this value, it indicates the HBA is ready to accept a new interface control command, although the transition to the previously selected state may not have occurred yet.																		

27	<b>ASP: Aggressive Slumber/Partial.</b> Read-write. Reset: 0. When set to 1, and ALPE==1, the HBA shall aggressively enter the Slumber state when it clears <a href="#">SATAx1[B,3]8</a> and <a href="#">SATAx1[B,3]4</a> is cleared or when it clears the <a href="#">SATAx1[B,3]4</a> register and <a href="#">SATAx1[B,3]8</a> is cleared. When cleared, and ALPE==1, the HBA shall aggressively enter the Partial state when it clears the <a href="#">SATAx1[B,3]8</a> register and the <a href="#">SATAx1[B,3]4</a> register is cleared or when it clears the <a href="#">SATAx1[B,3]4</a> register and <a href="#">SATAx1[B,3]8</a> is cleared. See <a href="#">SATAx00[SALP]</a> .
26	<b>ALPE: Aggressive Link Power Management Enable.</b> Read-write. Reset: 0. 1=The HBA shall aggressively enter a lower link power state (Partial or Slumber) based upon the setting of the ASP bit. See <a href="#">SATAx00[SALP]</a> .
25	<b>DLAE: Drive LED on ATAPI Enable.</b> Read-write. Reset: 0. 1=The HBA shall drive the LED pin active for commands regardless of the state of bit ATAPI. 0=The HBA shall only drive the LED pin active for commands if bit ATAPI==0.
24	<b>ATAPI: Device is ATAPI.</b> Read-write. Reset: 0. 1=The connected device is an ATAPI device. This bit is used by the HBA to control whether or not to generate the desktop LED when commands are active.
23	Reserved.
22	<b>FBSCP: FIS-based Switching Capable Port.</b> Read-only. Reset: 1. 1=Indicates that this port supports Port Multiplier FIS-based switching.
21	<b>ESP: External SATA Port.</b> Read-only. Reset: 0. 0=Indicates that this port's signal connector is not externally accessible on a signal only connector. ESP is mutually exclusive with the HPCP bit in this register.
20	<b>CPD: Cold Presence Detection.</b> Read-only. Reset: 0. 0=The platform does not support cold presence detection on this port.
19	<b>MPSP: Mechanical Presence Switch Attached to Port.</b> Read-only. Reset: 0. 0=The platform does not support a mechanical presence switch attached to this port.
18	<b>HPCP: Hot Plug Capable Port.</b> Read-only. Reset: 0. 0=Indicates that this port's signal and power connectors are not externally accessible via a joint signal and power connector.
17	<b>PMA: Port Multiplier Attached.</b> Read-write. Reset: 0. 1=A Port Multiplier is attached to the HBA for this port. 0=A Port Multiplier is not attached to the HBA for this port. Software is responsible for detecting whether a Port Multiplier is present; hardware does not auto-detect the presence of a Port Multiplier.
16	<b>CPS: Cold Presence State.</b> Read-only. Reset: 0. The CPS bit reports whether a device is currently detected on this port via cold presence detection. 1=The HBA detects via cold presence that a device is attached to this port. 0=The HBA detects via cold presence that there is no device attached to this port.
15	<b>CR: Command List Running.</b> Read-only. Reset: 0. 1=The command list DMA engine for the port is running.
14	<b>FR: FIS Receive Running.</b> Read-only. Reset: 0. 1=The FIS Receive DMA engine for the port is running.
13	<b>MPSS: Mechanical Presence Switch State.</b> Read-only. Reset: 1. This bit reports the state of a mechanical presence switch attached to this port. If <a href="#">SATAx00[SMPS]</a> ==1 and the mechanical presence switch is closed then this bit is cleared to 0. If <a href="#">SATAx00[SMPS]</a> ==1 and the mechanical presence switch is open then this bit is set to 1. If <a href="#">SATAx00[SMPS]</a> ==0 then this bit is cleared to 0. Software should only use this bit if (( <a href="#">SATAx00[SMPS]</a> ==1) && ( <a href="#">SATAx1[9,1]8[MPSP]</a> ==1)).

12:8	<b>CCS: Current Command Slot.</b> Read-only; updated-by-hardware. Reset: 0. This field is valid if <a href="#">SATAx1[9,1]8[ST]==1</a> and shall be set to the command slot value of the command that is currently being issued by the HBA. When <a href="#">SATAx1[9,1]8[ST]</a> transitions from 1 to 0, this field shall be reset to 0. After <a href="#">SATAx1[9,1]8[ST]</a> transitions from 0 to 1, the highest priority slot to issue from next is command slot 0. After the first command has been issued, the highest priority slot to issue from next is CCS+1. For example, after the HBA has issued its first command, if CCS==0 and <a href="#">SATAx1[B,3]8==3</a> , the next command that will be issued is from command slot 1.
7:5	Reserved.
4	<b>FRE: FIS Receive Enable.</b> Read-write. Reset: 0. 1=The HBA may post the received FISes into the FIS receive area pointed to by <a href="#">SATAx1[8,0]8 [Port 1,0 FIS Base Address (PxFB)]</a> (and <a href="#">SATAx1[8,0]C [Port 1,0 FIS Base Address Upper (PxFBU)]</a> for 64-bit HBAs). 0=Received FISes are not accepted by the HBA, except for the first D2H register FIS after the initialization sequence, and no FISes are posted to the FIS receive area. System software must not set this bit until <a href="#">SATAx1[8,0]8</a> (and <a href="#">SATAx1[8,0]C</a> for 64-bit) have been programmed with a valid pointer to the FIS receive area. If software wishes to move the base, this bit must first be cleared, and software must wait for the FR bit in this register to be cleared.
3	<b>CLO: Command List Override.</b> Read; write-1-only; cleared-by-hardware. Reset: 0. Setting this bit to 1 causes <a href="#">SATAx1[A,2]0[STS[7], STS[3]]</a> to be cleared to 0. This allows a software reset to be transmitted to the device regardless of whether the BSY and DRQ bits are still set in the <a href="#">SATAx1[A,2]0[STS]</a> register. The HBA sets this bit to 0 when <a href="#">SATAx1[A,2]0[STS[7], STS[3]]</a> have been cleared to 0. This bit shall only be set to 1 immediately prior to setting <a href="#">SATAx1[9,1]8[ST]</a> to 1 from a previous value of 0. Setting this bit to 1 at any other time is not supported and will result in indeterminate behavior. Software must wait for CLO to be cleared to 0 before setting <a href="#">SATAx1[9,1]8[ST]</a> to 1.
2	<b>POD: Power On Device.</b> Read-only. Reset: 1. 1=HBAs does not support cold presence detect and the HBA sets the state of a pin on the HBA to 1 so that it may be used to provide power to a cold-presence detectable port.
1	<b>SUD: Spin-Up Device.</b> Read-only. Reset: 1. 1=HBAs does not support staggered spin-up.
0	<b>ST: Start.</b> Read-write. Reset: 0. 1=The HBA may process the command list. 0=The HBA may not process the command list. Whenever this bit is changed from 0 to 1, the HBA starts processing the command list at entry 0. Whenever this bit is changed from a 1 to a 0, <a href="#">SATAx1[B,3]8</a> is cleared by the HBA upon the HBA putting the controller into an idle state. This bit will only be set to 1 by software after <a href="#">SATAx1[9,1]8[FRE]</a> has been set to 1.

**SATAx1[A,2]0 Port 1,0 Task File Data (PxTFD)**

Bits	Description												
31:16	Reserved.												
15:8	<b>ERR: Error.</b> Read-only. Reset: 0. Contains the latest copy of the task file error register.												
7:0	<b>STS: Status.</b> Read-only; updated-by-hardware. Reset: 7Fh. Contains the latest copy of the task file status register. Fields of note in this register that affect AHCI hardware operation are: <table border="0" style="margin-left: 20px;"> <tr> <td style="padding-right: 20px;"><u>Bit</u></td> <td><u>Definition</u></td> </tr> <tr> <td>[7]</td> <td>BSY. 1=Indicates the interface is busy</td> </tr> <tr> <td>[6:4]</td> <td>CS. Command specific</td> </tr> <tr> <td>[3]</td> <td>DRQ. 1=Indicates a data transfer is requested</td> </tr> <tr> <td>[2:1]</td> <td>CS. Command specific</td> </tr> <tr> <td>[0]</td> <td>ERR. 1=Indicates an error during the transfer</td> </tr> </table>	<u>Bit</u>	<u>Definition</u>	[7]	BSY. 1=Indicates the interface is busy	[6:4]	CS. Command specific	[3]	DRQ. 1=Indicates a data transfer is requested	[2:1]	CS. Command specific	[0]	ERR. 1=Indicates an error during the transfer
<u>Bit</u>	<u>Definition</u>												
[7]	BSY. 1=Indicates the interface is busy												
[6:4]	CS. Command specific												
[3]	DRQ. 1=Indicates a data transfer is requested												
[2:1]	CS. Command specific												
[0]	ERR. 1=Indicates an error during the transfer												

**SATAx1[A,2]4 Port 1,0 Signature (PxSIG)**

Bits	Description										
31:0	<p><b>SIG: Signature.</b> Read-write. Reset: FFFF_FFFFh. Contains the signature received from a device on the first D2H Register FIS. The bit order is as following:</p> <table border="0"> <thead> <tr> <th><u>Bit</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>[31:24]</td> <td>LBA High Register</td> </tr> <tr> <td>[23:16]</td> <td>LBA Mid Register</td> </tr> <tr> <td>[15:08]</td> <td>LBA Low Register</td> </tr> <tr> <td>[07:00]</td> <td>Sector Count Register</td> </tr> </tbody> </table> <p>The register is updated once after each reset sequence.</p>	<u>Bit</u>	<u>Definition</u>	[31:24]	LBA High Register	[23:16]	LBA Mid Register	[15:08]	LBA Low Register	[07:00]	Sector Count Register
<u>Bit</u>	<u>Definition</u>										
[31:24]	LBA High Register										
[23:16]	LBA Mid Register										
[15:08]	LBA Low Register										
[07:00]	Sector Count Register										

**SATAx1[A,2]8 Port 1,0 Serial ATA Status (PxSSTS)**

Bits	Description																		
31:12	Reserved.																		
11:8	<p><b>IPM: Interface Power Management.</b> Read-only; updated-by-hardware. Reset: 0. Indicates the current interface state.</p> <table border="0"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Device not present or communication not established</td> </tr> <tr> <td>1h</td> <td>Interface in Active state</td> </tr> <tr> <td>2h</td> <td>Interface in Partial power management state</td> </tr> <tr> <td>5h-3h</td> <td>Reserved</td> </tr> <tr> <td>6h</td> <td>Interface in Slumber power management state</td> </tr> <tr> <td>7h</td> <td>Reserved</td> </tr> <tr> <td>8h</td> <td>Interface in DevSleep power management state</td> </tr> <tr> <td>Fh-9h</td> <td>Reserved</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	0h	Device not present or communication not established	1h	Interface in Active state	2h	Interface in Partial power management state	5h-3h	Reserved	6h	Interface in Slumber power management state	7h	Reserved	8h	Interface in DevSleep power management state	Fh-9h	Reserved
<u>Bits</u>	<u>Definition</u>																		
0h	Device not present or communication not established																		
1h	Interface in Active state																		
2h	Interface in Partial power management state																		
5h-3h	Reserved																		
6h	Interface in Slumber power management state																		
7h	Reserved																		
8h	Interface in DevSleep power management state																		
Fh-9h	Reserved																		
7:4	<p><b>SPD: Current Interface Speed.</b> Read-only; updated-by-hardware. Reset: 0. Indicates the negotiated interface communication speed.</p> <table border="0"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Device not present or communication not established</td> </tr> <tr> <td>1h</td> <td>Generation 1 communication rate negotiated</td> </tr> <tr> <td>2h</td> <td>Generation 2 communication rate negotiated</td> </tr> <tr> <td>3h</td> <td>Generation 3 communication rate negotiated</td> </tr> <tr> <td>Fh-4h</td> <td>Reserved</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	0h	Device not present or communication not established	1h	Generation 1 communication rate negotiated	2h	Generation 2 communication rate negotiated	3h	Generation 3 communication rate negotiated	Fh-4h	Reserved						
<u>Bits</u>	<u>Definition</u>																		
0h	Device not present or communication not established																		
1h	Generation 1 communication rate negotiated																		
2h	Generation 2 communication rate negotiated																		
3h	Generation 3 communication rate negotiated																		
Fh-4h	Reserved																		
3:0	<p><b>DET: Device Detection.</b> Read-only; updated-by-hardware. Reset: 0. Indicates the interface device detection and PHY state.</p> <table border="0"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>No device detected and PHY communication not established</td> </tr> <tr> <td>1h</td> <td>Device presence detected but PHY communication not established</td> </tr> <tr> <td>2h</td> <td>Reserved.</td> </tr> <tr> <td>3h</td> <td>Device presence detected and PHY communication established</td> </tr> <tr> <td>4h</td> <td>PHY in offline mode as a result of the interface being disabled or running in a BIST loopback mode</td> </tr> <tr> <td>Fh-5h</td> <td>Reserved</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	0h	No device detected and PHY communication not established	1h	Device presence detected but PHY communication not established	2h	Reserved.	3h	Device presence detected and PHY communication established	4h	PHY in offline mode as a result of the interface being disabled or running in a BIST loopback mode	Fh-5h	Reserved				
<u>Bits</u>	<u>Definition</u>																		
0h	No device detected and PHY communication not established																		
1h	Device presence detected but PHY communication not established																		
2h	Reserved.																		
3h	Device presence detected and PHY communication established																		
4h	PHY in offline mode as a result of the interface being disabled or running in a BIST loopback mode																		
Fh-5h	Reserved																		



**SATAx1[A,2]C Port 1,0 Serial ATA Control (PxSCTL)**

Bits	Description																				
31:20	Reserved.																				
19:16	<b>PMP: Port Multiplier Port.</b> Read-only. Reset: 0. This field is not used by AHCI.																				
15:12	<b>SPM: Select Power Management.</b> Read-only. Reset: 0. This field is not used by AHCI.																				
11:8	<p><b>IPM: Interface Power Management Transitions Allowed.</b> Read-write. Reset: 0. Indicates which power states the HBA is allowed to transition to. If an interface power management state is disabled, the HBA is not allowed to initiate that state and the HBA must respond with power management denial to any request from the device to enter that state.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>No interface restrictions</td> </tr> <tr> <td>1h</td> <td>Disable transitions to the Partial state.</td> </tr> <tr> <td>2h</td> <td>Disable transitions to the Slumber state.</td> </tr> <tr> <td>3h</td> <td>Disable transitions to both Partial and Slumber states.</td> </tr> <tr> <td>4h</td> <td>Disable transitions to the DevSleep power management state.</td> </tr> <tr> <td>5h</td> <td>Disable transitions to the Partial and DevSleep power management states.</td> </tr> <tr> <td>6h</td> <td>Disable transitions to the Slumber and DevSleep power management states.</td> </tr> <tr> <td>7h</td> <td>Disable transitions to the Partial, Slumber and DevSleep power management states.</td> </tr> <tr> <td>Fh-8h</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Definition	0h	No interface restrictions	1h	Disable transitions to the Partial state.	2h	Disable transitions to the Slumber state.	3h	Disable transitions to both Partial and Slumber states.	4h	Disable transitions to the DevSleep power management state.	5h	Disable transitions to the Partial and DevSleep power management states.	6h	Disable transitions to the Slumber and DevSleep power management states.	7h	Disable transitions to the Partial, Slumber and DevSleep power management states.	Fh-8h	Reserved
Bits	Definition																				
0h	No interface restrictions																				
1h	Disable transitions to the Partial state.																				
2h	Disable transitions to the Slumber state.																				
3h	Disable transitions to both Partial and Slumber states.																				
4h	Disable transitions to the DevSleep power management state.																				
5h	Disable transitions to the Partial and DevSleep power management states.																				
6h	Disable transitions to the Slumber and DevSleep power management states.																				
7h	Disable transitions to the Partial, Slumber and DevSleep power management states.																				
Fh-8h	Reserved																				
7:4	<p><b>SPD: Speed Allowed.</b> Read-write. Reset: 0. Indicates the highest allowable speed of the interface.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>No speed negotiation restrictions</td> </tr> <tr> <td>1h</td> <td>Limit speed negotiation to Generation 1 communication rate</td> </tr> <tr> <td>2h</td> <td>Limit speed negotiation to a rate not greater than Generation 2 communication rate</td> </tr> <tr> <td>3h</td> <td>Limit speed negotiation to a rate not greater than Generation 3 communication rate</td> </tr> <tr> <td>Fh-4h</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Definition	0h	No speed negotiation restrictions	1h	Limit speed negotiation to Generation 1 communication rate	2h	Limit speed negotiation to a rate not greater than Generation 2 communication rate	3h	Limit speed negotiation to a rate not greater than Generation 3 communication rate	Fh-4h	Reserved								
Bits	Definition																				
0h	No speed negotiation restrictions																				
1h	Limit speed negotiation to Generation 1 communication rate																				
2h	Limit speed negotiation to a rate not greater than Generation 2 communication rate																				
3h	Limit speed negotiation to a rate not greater than Generation 3 communication rate																				
Fh-4h	Reserved																				
3:0	<p><b>DET: Device Detection Initialization.</b> Read-write. Reset: 0. Controls the HBA's device detection and interface initialization.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>No device detection or initialization action requested</td> </tr> <tr> <td>1h</td> <td>Perform interface communication initialization sequence to establish communication. This is functionally equivalent to a hard reset and results in the interface being reset and communications re-initialized. While this field is 1h, COMRESET is transmitted on the interface. Software should leave the DET field set to 1h for a minimum of 1 millisecond to ensure that a COMRESET is sent on the interface.</td> </tr> <tr> <td>3h-2h</td> <td>Reserved</td> </tr> <tr> <td>4h</td> <td>Disable the Serial ATA interface and put PHY in off-line mode</td> </tr> <tr> <td>Fh-5h</td> <td>Reserved</td> </tr> </tbody> </table> <p>This field may only be modified when <code>SATAx1[9,1]8[ST]==0</code>. Changing this field while <code>SATAx1[9,1]8[ST]==1</code> results in undefined behavior. When <code>SATAx1[9,1]8[ST]==1</code>, this field should have a value of 0.</p>	Bits	Definition	0h	No device detection or initialization action requested	1h	Perform interface communication initialization sequence to establish communication. This is functionally equivalent to a hard reset and results in the interface being reset and communications re-initialized. While this field is 1h, COMRESET is transmitted on the interface. Software should leave the DET field set to 1h for a minimum of 1 millisecond to ensure that a COMRESET is sent on the interface.	3h-2h	Reserved	4h	Disable the Serial ATA interface and put PHY in off-line mode	Fh-5h	Reserved								
Bits	Definition																				
0h	No device detection or initialization action requested																				
1h	Perform interface communication initialization sequence to establish communication. This is functionally equivalent to a hard reset and results in the interface being reset and communications re-initialized. While this field is 1h, COMRESET is transmitted on the interface. Software should leave the DET field set to 1h for a minimum of 1 millisecond to ensure that a COMRESET is sent on the interface.																				
3h-2h	Reserved																				
4h	Disable the Serial ATA interface and put PHY in off-line mode																				
Fh-5h	Reserved																				

**SATAx1[B,3]0 Port 1,0 Serial ATA Error(PxSERR)**

---

Bits	Description
------	-------------

31:16	<b>DIAG: Diagnostics.</b> Read; write-1-to-clear. Reset: 0. Contains diagnostic error information for use by diagnostic software in validating correct operation or isolating failure modes:																										
	<table border="1"> <thead> <tr> <th><u>Bit</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>[15:11]</td> <td>Reserved</td> </tr> <tr> <td>[10]</td> <td>Exchanged (X). Indicates a COMINIT signal was received. This bit is reflected in <a href="#">SATAx1[9,1]0[PCS]</a>.</td> </tr> <tr> <td>[9]</td> <td>Unknown FIS Type (F). Indicates that one or more FISs were received by the Transport layer with good CRC, but had a type field that was not recognized/known.</td> </tr> <tr> <td>[8]</td> <td>Transport state transition error (T). Indicates that an error has occurred in the transition from one state to another within the Transport layer since the last time this bit was cleared. This bit is always 0 in the current implementation.</td> </tr> <tr> <td>[7]</td> <td>Link Sequence Error (S). Indicates that one or more Link state machine error conditions was encountered. The Link Layer state machine defines the conditions under which the link layer detects an erroneous transition. This bit is always 0 in the current implementation.</td> </tr> <tr> <td>[6]</td> <td>Handshake Error (H). Indicates that one or more R_ERR handshake responses were received in response to frame transmission. Such errors may be the result of a CRC error detected by the recipient, a disparity or 8b/10b decoding error, or other error condition leading to a negative handshake on a transmitted frame.</td> </tr> <tr> <td>[5]</td> <td>CRC Error (C). Indicates that one or more CRC errors occurred with the Link Layer.</td> </tr> <tr> <td>[4]</td> <td>Disparity Error (D). This field is not used by AHCI. This bit is always 0 in the current implementation.</td> </tr> <tr> <td>[3]</td> <td>10B to 8B Decode Error (B). Indicates that one or more 10B to 8B decoding errors has occurred.</td> </tr> <tr> <td>[2]</td> <td>Comm Wake (W). Indicates that a Comm Wake signal was detected by the PHY.</td> </tr> <tr> <td>[1]</td> <td>Phy Internal Error (I). Indicates that the PHY detected some internal error. This bit is always 0 in the current implementation.</td> </tr> <tr> <td>[0]</td> <td>PhyRdy Change (N). Indicates that the PhyRdy signal changed state. This bit is reflected in the <a href="#">SATAx1[9,1]0[PRCS]</a> bit.</td> </tr> </tbody> </table>	<u>Bit</u>	<u>Definition</u>	[15:11]	Reserved	[10]	Exchanged (X). Indicates a COMINIT signal was received. This bit is reflected in <a href="#">SATAx1[9,1]0[PCS]</a> .	[9]	Unknown FIS Type (F). Indicates that one or more FISs were received by the Transport layer with good CRC, but had a type field that was not recognized/known.	[8]	Transport state transition error (T). Indicates that an error has occurred in the transition from one state to another within the Transport layer since the last time this bit was cleared. This bit is always 0 in the current implementation.	[7]	Link Sequence Error (S). Indicates that one or more Link state machine error conditions was encountered. The Link Layer state machine defines the conditions under which the link layer detects an erroneous transition. This bit is always 0 in the current implementation.	[6]	Handshake Error (H). Indicates that one or more R_ERR handshake responses were received in response to frame transmission. Such errors may be the result of a CRC error detected by the recipient, a disparity or 8b/10b decoding error, or other error condition leading to a negative handshake on a transmitted frame.	[5]	CRC Error (C). Indicates that one or more CRC errors occurred with the Link Layer.	[4]	Disparity Error (D). This field is not used by AHCI. This bit is always 0 in the current implementation.	[3]	10B to 8B Decode Error (B). Indicates that one or more 10B to 8B decoding errors has occurred.	[2]	Comm Wake (W). Indicates that a Comm Wake signal was detected by the PHY.	[1]	Phy Internal Error (I). Indicates that the PHY detected some internal error. This bit is always 0 in the current implementation.	[0]	PhyRdy Change (N). Indicates that the PhyRdy signal changed state. This bit is reflected in the <a href="#">SATAx1[9,1]0[PRCS]</a> bit.
<u>Bit</u>	<u>Definition</u>																										
[15:11]	Reserved																										
[10]	Exchanged (X). Indicates a COMINIT signal was received. This bit is reflected in <a href="#">SATAx1[9,1]0[PCS]</a> .																										
[9]	Unknown FIS Type (F). Indicates that one or more FISs were received by the Transport layer with good CRC, but had a type field that was not recognized/known.																										
[8]	Transport state transition error (T). Indicates that an error has occurred in the transition from one state to another within the Transport layer since the last time this bit was cleared. This bit is always 0 in the current implementation.																										
[7]	Link Sequence Error (S). Indicates that one or more Link state machine error conditions was encountered. The Link Layer state machine defines the conditions under which the link layer detects an erroneous transition. This bit is always 0 in the current implementation.																										
[6]	Handshake Error (H). Indicates that one or more R_ERR handshake responses were received in response to frame transmission. Such errors may be the result of a CRC error detected by the recipient, a disparity or 8b/10b decoding error, or other error condition leading to a negative handshake on a transmitted frame.																										
[5]	CRC Error (C). Indicates that one or more CRC errors occurred with the Link Layer.																										
[4]	Disparity Error (D). This field is not used by AHCI. This bit is always 0 in the current implementation.																										
[3]	10B to 8B Decode Error (B). Indicates that one or more 10B to 8B decoding errors has occurred.																										
[2]	Comm Wake (W). Indicates that a Comm Wake signal was detected by the PHY.																										
[1]	Phy Internal Error (I). Indicates that the PHY detected some internal error. This bit is always 0 in the current implementation.																										
[0]	PhyRdy Change (N). Indicates that the PhyRdy signal changed state. This bit is reflected in the <a href="#">SATAx1[9,1]0[PRCS]</a> bit.																										

15:0	<p><b>Err: Error.</b> Read; write-1-to-clear. Reset: 0. Contains error information for use by host software in determining the appropriate response to the error condition.</p> <table border="1"> <thead> <tr> <th style="text-align: left;"><u>Bit</u></th> <th style="text-align: left;"><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>[15:12]</td> <td>Reserved.</td> </tr> <tr> <td>[11]</td> <td>Internal Error (E). The host bus adapter experienced an internal error that caused the operation to fail and may have put the host bus adapter into an error state. The internal error may include a master or target abort when attempting to access system memory, an elasticity buffer overflow, a primitive mis-alignment, a synchronization FIFO overflow, and other internal error conditions. Typically, when an internal error occurs, a non-fatal or fatal status bit in <a href="#">SATAx1[9,1]0</a> register will also be set to give software guidance on the recovery mechanism required.</td> </tr> <tr> <td>[10]</td> <td>Protocol Error (P). A violation of the Serial ATA protocol was detected.</td> </tr> <tr> <td>[9]</td> <td>Persistent Communication or Data Integrity Error (C). A communication error that was not recovered occurred and is expected to be persistent. Persistent communication errors may arise from faulty interconnect with the device, from a device that has been removed or has failed, or a number of other causes.</td> </tr> <tr> <td>[8]</td> <td>Transient Data Integrity Error (T). A data integrity error occurred that was not recovered by the interface. This bit is set upon any error when a Data FIS is received, including reception FIFO overflow, CRC error, or 10b/8b decoding error.</td> </tr> <tr> <td>[7:2]</td> <td>Reserved.</td> </tr> <tr> <td>[1]</td> <td>Recovered Communications Error (M). Communications between the device and host was temporarily lost but was re-established. This can arise from a device temporarily being removed, from a temporary loss of PHY synchronization, or from other causes and may be derived from the PhyNRdy signal between the PHY and Link layers.</td> </tr> <tr> <td>[0]</td> <td>Recovered Data Integrity Error (I). A data integrity error occurred that was recovered by the interface through a retry operation or other recovery action. This bit is set upon any error when a Data FIS is received, including reception FIFO overflow, CRC error, or 10b/8b decoding error.</td> </tr> </tbody> </table>	<u>Bit</u>	<u>Definition</u>	[15:12]	Reserved.	[11]	Internal Error (E). The host bus adapter experienced an internal error that caused the operation to fail and may have put the host bus adapter into an error state. The internal error may include a master or target abort when attempting to access system memory, an elasticity buffer overflow, a primitive mis-alignment, a synchronization FIFO overflow, and other internal error conditions. Typically, when an internal error occurs, a non-fatal or fatal status bit in <a href="#">SATAx1[9,1]0</a> register will also be set to give software guidance on the recovery mechanism required.	[10]	Protocol Error (P). A violation of the Serial ATA protocol was detected.	[9]	Persistent Communication or Data Integrity Error (C). A communication error that was not recovered occurred and is expected to be persistent. Persistent communication errors may arise from faulty interconnect with the device, from a device that has been removed or has failed, or a number of other causes.	[8]	Transient Data Integrity Error (T). A data integrity error occurred that was not recovered by the interface. This bit is set upon any error when a Data FIS is received, including reception FIFO overflow, CRC error, or 10b/8b decoding error.	[7:2]	Reserved.	[1]	Recovered Communications Error (M). Communications between the device and host was temporarily lost but was re-established. This can arise from a device temporarily being removed, from a temporary loss of PHY synchronization, or from other causes and may be derived from the PhyNRdy signal between the PHY and Link layers.	[0]	Recovered Data Integrity Error (I). A data integrity error occurred that was recovered by the interface through a retry operation or other recovery action. This bit is set upon any error when a Data FIS is received, including reception FIFO overflow, CRC error, or 10b/8b decoding error.
<u>Bit</u>	<u>Definition</u>																		
[15:12]	Reserved.																		
[11]	Internal Error (E). The host bus adapter experienced an internal error that caused the operation to fail and may have put the host bus adapter into an error state. The internal error may include a master or target abort when attempting to access system memory, an elasticity buffer overflow, a primitive mis-alignment, a synchronization FIFO overflow, and other internal error conditions. Typically, when an internal error occurs, a non-fatal or fatal status bit in <a href="#">SATAx1[9,1]0</a> register will also be set to give software guidance on the recovery mechanism required.																		
[10]	Protocol Error (P). A violation of the Serial ATA protocol was detected.																		
[9]	Persistent Communication or Data Integrity Error (C). A communication error that was not recovered occurred and is expected to be persistent. Persistent communication errors may arise from faulty interconnect with the device, from a device that has been removed or has failed, or a number of other causes.																		
[8]	Transient Data Integrity Error (T). A data integrity error occurred that was not recovered by the interface. This bit is set upon any error when a Data FIS is received, including reception FIFO overflow, CRC error, or 10b/8b decoding error.																		
[7:2]	Reserved.																		
[1]	Recovered Communications Error (M). Communications between the device and host was temporarily lost but was re-established. This can arise from a device temporarily being removed, from a temporary loss of PHY synchronization, or from other causes and may be derived from the PhyNRdy signal between the PHY and Link layers.																		
[0]	Recovered Data Integrity Error (I). A data integrity error occurred that was recovered by the interface through a retry operation or other recovery action. This bit is set upon any error when a Data FIS is received, including reception FIFO overflow, CRC error, or 10b/8b decoding error.																		

**SATAx1[B,3]4 Port 1,0 Serial ATA Active (PxSACT)**

Bits	Description
31:0	<p><b>DS: Device Status.</b> Read; write-1-only; cleared-by-hardware. Reset: 0. This field is bit significant. Each bit corresponds to the TAG and command slot of a native queued command, where bit 0 corresponds to TAG 0 and command slot 0. This field is set by software prior to issuing a native queued command for a particular command slot. Prior to writing <a href="#">SATAx1[B,3]8</a> TAG to 1, software will set DS TAG to 1 to indicate that a command with that TAG is outstanding. The device clears bits in this field by sending a Set Device Bits FIS to the host. The HBA clears bits in this field that are set to 1 in the SActive field of the Set Device Bits FIS. The HBA only clears bits that correspond to native queued commands that have completed successfully.</p> <p>Software should only write to this field when <a href="#">SATAx1[9,1]8[ST]==1</a>. This field is cleared when <a href="#">SATAx1[9,1]8[ST]</a> is written from 1 to 0 by software. This field is not cleared by a COMRESET or a software reset.</p>

**SATAx1[B,3]8 Port 1,0 Command Issue (PxCI)**

Bits	Description
31:0	<b>CI: Commands Issued.</b> Read; write-1-only; cleared-by-hardware. Reset: 0. This field is bit significant. Each bit corresponds to a command slot, where bit 0 corresponds to command slot 0. This field is set by software to indicate to the HBA that a command has been built in system memory for a command slot and may be sent to the device. When the HBA receives a FIS which clears the BSY, DRQ, and ERR bits for the command, it clears the corresponding bit in this register for that command slot. Bits in this field shall only be set to 1 by software when <a href="#">SATAx1[9,1]8[ST]</a> is set to 1. This field is also cleared when <a href="#">SATAx1[9,1]8[ST]</a> is written from 1 to 0 by software.

**SATAx1[B,3]C Port 1,0 SNotification (PxSNTF)**

Bits	Description
31:16	Reserved.
15:0	<b>PMN: PM Notify.</b> Read; write-1-to-clear. Reset: 0. This field is bit significant. Each bit corresponds to a Port Multiplier (PM) port number, where bit 0 corresponds to PM port 0 and bit 15 corresponds to PM port 15. This field indicates whether a particular device with the corresponding PM Port number issued a Set Device Bits FIS to the host with the Notification bit set. Individual bits are cleared by software writing 1 to the corresponding bit positions. This field is reset to default on a HBA Reset, but it is not reset by COMRESET or software reset.

**SATAx1[C,4]0 Port 1,0 FIS-based Switching Control (PxFBS)**

Bits	Description
31:20	Reserved.
19:16	<b>DWE: Device With Error.</b> Read-only; updated-by-hardware. Reset: 0. Set by hardware to the value of the Port Multiplier port number of the device that experienced a fatal error condition. This field is only valid when <a href="#">SATAx1[C,4]0[SDE]</a> ==1.
15:12	<b>ADO: Active Device Optimization.</b> Read-only. . Reset: 2h. This register exposes the number of active devices that the FIS-based switching implementation has been optimized for. When there are more devices active than indicated in this field, throughput of concurrent traffic may degrade. For optimal performance, software should limit the number of active devices based on this value. The minimum value for this field shall be 2h, indicating that at least two devices may be active with high performance maintained.
11:8	<b>DEV: Device To Issue.</b> Read-write. Reset: 0. Set by software to the Port Multiplier port value of the next command to issue. This field enables hardware to know the port the command is to be issued to without fetching the command header. Software shall not issue commands to multiple Port Multiplier ports on the same write of <a href="#">SATAx1[B,3]8</a> register.
7:3	Reserved.

2	<b>SDE: Single Device Error.</b> Read-only. Reset: 0. 1=When a fatal error condition has occurred, hardware believes the error is localized to one device such that software's first error recovery step should be to utilize the <a href="#">SATAx1[C,4]0[DEC]</a> functionality. 0=When a fatal error condition has occurred, the error applies to the entire port and to clear the error <a href="#">SATAx1[9,1]8[ST]</a> shall be cleared to 0 by software. This bit is cleared on <a href="#">SATAx1[C,4]0[DEC]</a> being set to 1 or on <a href="#">SATAx1[9,1]8[ST]</a> being cleared to 0.
1	<b>DEC: Device Error Clear.</b> Read; write-1-only; cleared-by-hardware. Reset: 0. When set to 1 by software, the HBA shall clear the device-specific error condition and the HBA shall flush any commands outstanding for the device that experienced the error, including clearing <a href="#">SATAx1[B,3]8</a> and <a href="#">SATAx1[B,3]4</a> bits for that device to 0. When hardware has completed error recovery actions, hardware shall clear the bit to 0. Software shall only set this bit to 1 if (( <a href="#">SATAx1[C,4]0[EN]</a> ==1) && ( <a href="#">SATAx1[C,4]0[SDE]</a> ==1)).
0	<b>EN: Enable.</b> Read-write. Reset: 0. 1=A Port Multiplier is attached and the HBA shall use FIS-based switching to communicate with it. 0=FIS-based switching is not being used. Software shall only change the value of this bit when <a href="#">SATAx1[9,1]8[ST]</a> ==0.

**SATAx1[C,4]4 Port 1,0 Device Sleep (PxDEVSLP)**

Bits	Description												
31:29	Reserved.												
28:25	<b>DM: DITO Multiplier.</b> Read-only. Reset: 0. 0's based value that specifies the DITO multiplier that the HBA applies to the specified DITO value, effectively extending the range of DITO from 1 ms to 16383 ms. The HBA computes the total idle timeout as a product of DM and DITO (i.e. DITOfactual = DITO * DM).  <table style="width: 100%; border: none;"> <thead> <tr> <th style="text-align: center;"><u>Bits</u></th> <th style="text-align: center;"><u>Definition</u></th> <th style="text-align: center;"><u>Bits</u></th> <th style="text-align: center;"><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0h</td> <td style="text-align: center;">multiply 1</td> <td style="text-align: center;">Eh-2h</td> <td style="text-align: center;">multiply &lt;DM+1&gt;</td> </tr> <tr> <td style="text-align: center;">1h</td> <td style="text-align: center;">multiply 2</td> <td style="text-align: center;">Fh</td> <td style="text-align: center;">multiply 16</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	<u>Bits</u>	<u>Definition</u>	0h	multiply 1	Eh-2h	multiply <DM+1>	1h	multiply 2	Fh	multiply 16
<u>Bits</u>	<u>Definition</u>	<u>Bits</u>	<u>Definition</u>										
0h	multiply 1	Eh-2h	multiply <DM+1>										
1h	multiply 2	Fh	multiply 16										
24:15	<b>DITO: Device Sleep Idle Timeout.</b> IF (( <a href="#">SATAx24[SDS]</a> ==0)    ( <a href="#">SATAx24[SADM]</a> ==0)) THEN Read-only. ELSE Read-write. ENDIF. Reset: 0. This field specifies the amount of the time (in approximate 1 ms granularity) that the HBA shall wait before driving the DEVSLP signal. Hardware reloads its port specific Device Sleep timer with this value each time the port transitions out of the DEVSLP state. For example: from DevSleep to Active or <a href="#">SATAx1[C,4]4[ADSE]</a> transitions from 0 to 1. Software shall only set this value when <a href="#">SATAx1[9,1]8[ST]</a> ==0 and <a href="#">SATAx1[C,4]4[ADSE]</a> ==0.												
14:10	<b>MDAT: Minimum Device Sleep Assertion Time.</b> IF ( <a href="#">SATAx24[SDS]</a> ==0) THEN Read-only. ELSE Read-write. ENDIF. Reset: 0. This field specifies the minimum amount of time (in 1 ms granularity) that the HBA must assert the DEVSLP signal before it may be de-asserted. The nominal value is 10 ms and the minimum is 1 ms depending on device identification information. Software shall only set this value when <a href="#">SATAx1[9,1]8[ST]</a> ==0, <a href="#">SATAx1[C,4]4[ADSE]</a> ==0 and prior to setting <a href="#">SATAx1[9,1]8[ICC]</a> to 8h.												
9:2	<b>DETO: Device Sleep Exit Timeout.</b> IF ( <a href="#">SATAx24[SDS]</a> ==0) THEN Read-only. ELSE Read-write. ENDIF. Reset: 0. This field specifies the maximum duration (in approximate 1 ms granularity) from DEVSLP de-assertion until the device is ready to accept OOB. The nominal value is 20 ms while the max value is 255 ms depending on device identification information. Software shall only set this value when <a href="#">SATAx1[9,1]8[ST]</a> ==0, <a href="#">SATAx1[C,4]4[ADSE]</a> ==0 and prior to setting <a href="#">SATAx1[9,1]8[ICC]</a> to 8h.												

1	<p><b>DSP: Device Sleep Present.</b> Read-write. Reset: 0. 1=The platform supports Device Sleep on this port. 0=The platform does not support Device Sleep on this port.                  This bit may only be set to 1 if <code>SATAx24[SDS]==1</code>. DSP is mutually exclusive with the <code>SATAx1[9,1]8[HPCP]</code> bit and <code>SATAx1[9,1]8[ESP]</code> bit.</p>
0	<p><b>ADSE: Aggressive Device Sleep Enable.</b> IF ((<code>SATAx24[SADM]==1</code>) &amp;&amp; (<code>SATAx24[SDS]==1</code>)) THEN Read-write. ELSE Read-only. ENDIF. Reset: 0. 0=The HBA does not enter DevSleep unless software directed via <code>SATAx1[9,1]8[ICC]</code>. 1=The HBA shall assert the DEVSLP signal after the port has been idle (<code>SATAx1[B,3]8==0</code> and <code>SATAx1[B,3]4==0</code>) for the amount of time specified by the <code>SATAx1[C,4]4[DITO]</code>; If (<code>SATAx24[DESO]== 1</code>), the HBA shall assert the DEVSLP signal after the port has been idle for the amount of time specified by <code>SATAx1[C,4]4[DITO]</code> and the interface is in Slumber state (<code>SATAx1[A,2]8[IPM]==6h</code>).                  This bit shall only be set to 1 if <code>SATAx1[C,4]4[DSP]==1</code>.</p>

**3.25.3.3.3 Enclosure Buffer Management Registers**

The Enclosure Buffer Management registers range from (`D11F0x24 [AHCI Base Address (BAR5)] + SATAx1C[OFST]*4 + 00h`) to (`D11F0x24 [AHCI Base Address (BAR5)] + SATAx1C[OFST]*4 + FFh`) within the memory mapped space.

**SATA\_EMx00 Message Header**

Bits	Description
31:28	Reserved.
27:24	<b>MTYPE: Message Type.</b> Read-write. Reset: 0.
23:16	<b>DSIZE: Data Size.</b> Read-write. Reset: 0.
15:8	<b>MSIZE: Message Size.</b> Read-write. Reset: 0.
7:0	Reserved.

**SATA\_EMx04 Write SGPIO Register Request (I)**

Bits	Description																
31:24	<p><b>REG_INDX: Register Index.</b> Read-write. Reset: 0. Specifies the index of the first register in the bank to write. Currently supported index range is:</p> <table border="1"> <thead> <tr> <th><u>REG TYPE</u></th> <th><u>REG INDEX</u></th> </tr> </thead> <tbody> <tr> <td>00h</td> <td>0-1</td> </tr> <tr> <td>03h</td> <td>0</td> </tr> <tr> <td>04h</td> <td>0-7</td> </tr> <tr> <td>0Ch</td> <td>0</td> </tr> </tbody> </table> <p>It is software's responsibility to make sure the index of which the first register it attempts to access is valid and supported, as hardware does not provide a mechanism to return a status to software indicating that the current operation fails. An out-of-range index for a specific type of register will be treated as don't care and none of the data for such index will be written.</p>	<u>REG TYPE</u>	<u>REG INDEX</u>	00h	0-1	03h	0	04h	0-7	0Ch	0						
<u>REG TYPE</u>	<u>REG INDEX</u>																
00h	0-1																
03h	0																
04h	0-7																
0Ch	0																
23:16	<p><b>REG_TYPE: Register Type.</b> Read-write. Reset: 0. Specifies the bank of registers to write. Currently supported Register Types are:</p> <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>SGPIO Configuration</td> </tr> <tr> <td>2h-1h</td> <td>Reserved</td> </tr> <tr> <td>3h</td> <td>Transmit</td> </tr> <tr> <td>4h</td> <td>General Purpose Transmit</td> </tr> <tr> <td>Bh-5h</td> <td>Reserved</td> </tr> <tr> <td>Ch</td> <td>AMD Configuration</td> </tr> <tr> <td>FFh-Dh</td> <td>Reserved</td> </tr> </tbody> </table> <p>It is software's responsibility to make sure the type of registers that it attempts to access is valid and supported, as hardware does not provide a mechanism to return a status to software indicating that the current operation fails.</p>	<u>Bits</u>	<u>Definition</u>	0h	SGPIO Configuration	2h-1h	Reserved	3h	Transmit	4h	General Purpose Transmit	Bh-5h	Reserved	Ch	AMD Configuration	FFh-Dh	Reserved
<u>Bits</u>	<u>Definition</u>																
0h	SGPIO Configuration																
2h-1h	Reserved																
3h	Transmit																
4h	General Purpose Transmit																
Bh-5h	Reserved																
Ch	AMD Configuration																
FFh-Dh	Reserved																
15:8	<b>FUNC: Function.</b> Read-write. Reset: 0.																
7:0	<b>FRAME_TYPE: Frame Type.</b> Read-write. Reset: 0.																

**SATA\_EMx08 Write SGPIO Register Request (II)**

Bits	Description										
31:8	Reserved.										
7:0	<p><b>REG_CNT: Register Count.</b> Read-write. Reset: 0. Specifies the number of registers starting with the specified index to write. Currently supported register count range is:</p> <table border="1"> <thead> <tr> <th><u>REG TYPE</u></th> <th><u>REG CNT</u></th> </tr> </thead> <tbody> <tr> <td>00h</td> <td>1-2</td> </tr> <tr> <td>03h</td> <td>1</td> </tr> <tr> <td>04h</td> <td>1-8</td> </tr> <tr> <td>0Ch</td> <td>1</td> </tr> </tbody> </table> <p>It is software's responsibility to make sure the register count is valid and supported, as hardware does not provide a mechanism to return a status to software indicating that the current operation fails. An out-of-range register count for a specific type of register will be treated as don't care and none of the data for such register count will be written.</p>	<u>REG TYPE</u>	<u>REG CNT</u>	00h	1-2	03h	1	04h	1-8	0Ch	1
<u>REG TYPE</u>	<u>REG CNT</u>										
00h	1-2										
03h	1										
04h	1-8										
0Ch	1										



**SATA\_EMx0C Write SGPIO Register Request (III)**

Bits	Description
255:0	<b>DataRegister[7:0]: Data Register [7:0]</b> . Read-write. Reset: 0. This field contains 256 bits of data registers ranging from (D11F0x24 [AHCI Base Address (BAR5)] + SATAx1C[OFST] * 4 + 0Ch) to (D11F0x24 [AHCI Base Address (BAR5)] + SATAx1C[OFST] * 4 + 2Bh). Once <a href="#">SATA_EMx04[REG_TYPE]</a> , <a href="#">SATA_EMx04[REG_INDX]</a> , and <a href="#">SATA_EMx08[REG_CNT]</a> are specified, the values in this field will be written to the internal copy of SGPIO registers.

### 3.25.4 USB Controllers

#### 3.25.4.1 USB 1.1 (OHCI)

The OHCI controllers can be disabled by programming [PMxEF](#).

##### 3.25.4.1.1 Devices 16h, 13h, 12h Function 0 (OHCI) 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[16,13,12]F0x00 Device/Vendor ID

Bits	Description
31:16	<b>DeviceID: device ID.</b> Value: 7807h. This 16-bit field is assigned by the device manufacturer and identifies the type of device.
15:0	<b>VendorID: vendor ID.</b> Value: 1022h.

#### D[16,13,12]F0x04 Status/Command

Bits	Description
31	<b>ParityErrorDetected: parity error detected.</b> Read; set-by-hardware; write-1-to-clear. Reset: 0. Set by the device when it detects a parity error. This is not affected by ParityErrorEn.
30	<b>SignaledSystemError: signaled system error.</b> Read; set-by-hardware; write-1-to-clear. Reset: 0. Set when the device asserts SERR#.
29	<b>ReceivedMasterAbort: received master abort.</b> Read; set-by-hardware; write-1-to-clear. Reset: 0. Set by a master device when its transaction (except for Special Cycle) is terminated with master-abort.
28	<b>ReceivedTargetAbort: received target abort.</b> Read; set-by-hardware; write-1-to-clear. Reset: 0. Set by a master device when its transaction is terminated with target-abort.
27	<b>SignaledTargetAbort: signaled target abort.</b> Read; set-by-hardware; write-1-to-clear. Reset: 0. Set by a target device when it terminates a transaction with target-abort.
26:25	<b>DevSelTiming: DEVSEL# timing.</b> Value: 01b. 01b=Medium timing.
24	<b>MasterDataParityError: master data parity error.</b> Read; set-by-hardware; write-1-to-clear. Reset: 0.
23	<b>FastB2BCap: fast back-to-back capable.</b> Value: 1. 1=Fast back-to-back capable.
22	Reserved.
21	<b>PCI66En: 66 MHz capable.</b> Value: 1. 1=66 MHz capable.
20	<b>CapList: capabilities list.</b> Value: 1. 1=D[16,13,12]F0x34 [Capabilities Pointer] points to a linked list of capabilities.
19	<b>InterruptStatus: interrupt status.</b> Read-only. Reset: 0. 1=An INTx interrupt Message is pending in the device.
18:11	Reserved.
10	<b>InterruptDis: interrupt disable.</b> Read-write. Reset: 0. 0=Enable the assertion of the device/function's INTx# signal. 1=Disable the assertion of the device/function's INTx# signal.

9	<b>FastB2BEn: fast back-to-back enable.</b> Read-write. Reset: 0. 0=Only fast back-to-back transactions to the same agent are allowed. 1=The master is allowed to generate fast back-to-back transactions to different agents.
8	<b>SerrEn: system error enable.</b> Read-write. Reset: 0. 0=Disable the SERR# driver. 1=Enable the SERR# driver. Address parity errors are reported only if both SerrEn and ParityErrorEn are 1.
7	Reserved.
6	<b>ParityErrorEn: parity error response enable.</b> Read-write. Reset: 0. 0=The device sets <b>D[16,13,12]F0x04[ParityErrorDetected]</b> when an error is detected, but continues normal operations without asserting PERR#. 1=The device takes its normal action when a parity error is detected.
5	<b>VgaPaletteRegisterAccesses: VGA palette register accesses.</b> Value: 0. 0=Indicates the device should treat palette write accesses like all other accesses.
4	<b>MemWriteInvalidateEn: memory write and invalidate command enable.</b> Read-write. Reset: 0. 0=Memory write must be used. 1=Masters may use the memory write and invalidate command.
3	<b>SpecialCycle: special cycle.</b> Value: 0. 0=No Special Cycle support.
2	<b>BusMasterEn: bus master enable.</b> Read-write. Reset: 0. 0=Disable the device from generating PCI accesses. 1=Allow the device to behave as a bus master.
1	<b>MemAccessEn: memory access enable.</b> Read-write. Reset: 0. Specifies whether memory accesses to this device are enabled. 1=Enabled. 0=Disabled.
0	<b>IOAccessEn: IO access enable.</b> Read-write. Reset: 0. Specifies whether IO accesses to this device are enabled. 1=Enabled. 0=Disabled.

#### **D[16,13,12]F0x08 Class Code/Revision ID**

Bits	Description
31:24	<b>BC: base class.</b> Value: 0Ch. 0Ch=Device is a Serial Bus Controller.
23:16	<b>SC: sub class.</b> Value: 03h. 03h=Device is a Universal Serial Bus.
15:8	<b>PI: programming interface.</b> Value: 10h. 10h=Device is an OpenHCI Host Controller.
7:0	<b>RevisionID: revision ID.</b> Value: 39h.

#### **D[16,13,12]F0x0C Miscellaneous**

Bits	Description
31:24	<b>Bist.</b> Value: 00h. 00h=BIST is not supported.
23:16	<b>HeaderType: header type.</b> Value: 80h. 80h=This device contains multiple functions.
15:10	<b>LatencyTimer[7:2]: latency timer.</b> Read-write. Reset: 0. This field specifies, in units of PCI bus clocks, the value of the latency timer for this PCI bus master.
9:8	<b>LatencyTimer[1:0].</b> Value: 0. See LatencyTimer[7:2]. LatencyTimer[1:0] are hard-wired to 00b, resulting in a timer granularity of at least four clocks.
7:0	<b>CacheLineSize: cache line size.</b> Read-write. Reset: 0. Specifies the system cache line size in units of doublewords.

**D[16,13,12]F0x10 OHCI Base Address**

Bits	Description
31:12	<b>Bar: base address.</b> Read-write. Reset: 0. Specifies the upper 20 bits of the 32-bit base address for the OHCI memory mapped IO registers. See 3.25.4.1.2 [OHCI Memory Mapped IO Registers].
11:4	Reserved.
3	<b>Pref: prefetchable.</b> Value: 0. 0=Non-prefetchable memory region.
2:1	<b>Type: base address register type.</b> Value: 00b. 00b=32-bit base address register.
0	<b>MemSpace: memory space type.</b> Value: 0. 0=Memory mapped base address.

**D[16,13,12]F0x2C Subsystem and Subvendor ID**

Bits	Description
31:16	<b>SubsystemID: subsystem ID.</b> Write-once. Reset: 7807h.
15:0	<b>SubsystemVendorID: subsystem vendor ID.</b> Write-once. Reset: 1022h.

**D[16,13,12]F0x34 Capabilities Pointer**

Bits	Description
31:8	Reserved.
7:0	<b>CapPtr: capabilities pointer.</b> Value: D0h. Address of the first element in the capabilities linked list.

**D[16,13,12]F0x3C Interrupt Line**

Bits	Description
31:24	<b>MaxLat.</b> Value: 00h.
23:16	<b>MinGnt.</b> Value: 00h.
15:8	<b>InterruptPin: interrupt pin.</b> Read-only. Reset: 01h. 01h=INTA#.
7:0	<b>InterruptLine: interrupt line routing.</b> Read-write. Reset: 0. POST software will write the interrupt routing information into this register as it initializes and configures the system.

**D[16,13,12]F0x40 Miscellaneous 2**

Bits	Description												
31:20	Reserved. Reset: F00h.												
19:16	<b>PortDisable.</b> Read; write-once; write-1-only. Reset: 0h. 1=The corresponding port is disabled. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Bit</th> <th>Definition</th> <th>Bit</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>[0]</td> <td>Port 0</td> <td>[2]</td> <td>Port 2</td> </tr> <tr> <td>[1]</td> <td>Port 1</td> <td>[3]</td> <td>Port 3</td> </tr> </tbody> </table>	Bit	Definition	Bit	Definition	[0]	Port 0	[2]	Port 2	[1]	Port 1	[3]	Port 3
Bit	Definition	Bit	Definition										
[0]	Port 0	[2]	Port 2										
[1]	Port 1	[3]	Port 3										
15:9	Reserved.												

8	<b>MsiDisable: MSI disable.</b> Read-write. Reset: 1. 1=MSI capability is disabled.
7:0	<b>TrdyTimer: TRDY timer.</b> Read-write. Reset: 80h. Target Ready timeout for non-responding target.

**D[16,13,12]F0x44 USB Battery Charger**

Bits	Description
31:0	Reserved.

**D[16,13,12]F0x48 Port Force Reset**

Bits	Description								
31:8	Reserved.								
7:6	<b>Port3ForceResetEn: port 3 force reset enable.</b> Read-write. Reset: 0. See: Port0ForceResetEn.								
5:4	<b>Port2ForceResetEn: port 2 force reset enable.</b> Read-write. Reset: 0. See: Port0ForceResetEn.								
3:2	<b>Port1ForceResetEn: port 1 force reset enable.</b> Read-write. Reset: 0. See: Port0ForceResetEn.								
1:0	<b>Port0ForceResetEn: port 0 force reset enable.</b> Read-write. Reset: 0. Specifies whether the port is held in reset when the system is in S3, S4, or S5. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>x0b</td> <td>Port is not held in reset when the system is in S3, S4, or S5.</td> </tr> <tr> <td>01b</td> <td>The host controller will hold the port in reset when the system is in S3, S4, or S5 and the port is not connected.</td> </tr> <tr> <td>11b</td> <td>The host controller will hold the port in reset when the system is in S3, S4, or S5 regardless of the port connection status.</td> </tr> </tbody> </table>	Bits	Definition	x0b	Port is not held in reset when the system is in S3, S4, or S5.	01b	The host controller will hold the port in reset when the system is in S3, S4, or S5 and the port is not connected.	11b	The host controller will hold the port in reset when the system is in S3, S4, or S5 regardless of the port connection status.
Bits	Definition								
x0b	Port is not held in reset when the system is in S3, S4, or S5.								
01b	The host controller will hold the port in reset when the system is in S3, S4, or S5 and the port is not connected.								
11b	The host controller will hold the port in reset when the system is in S3, S4, or S5 regardless of the port connection status.								

**D[16,13,12]F0x50 OHCI Misc Control**

This register is shared among all the OHCI controllers if (D[16,13,12]F0x50[ShareRegisterEnable]==1).

Bits	Description
31	<b>PmeMergeDisable: PME merge disable.</b> Read-write. Reset: 1. 1=Disable merging PMEs from all USB controllers.
30:29	Reserved.
28	<b>ShareRegisterEnable: share register enable.</b> Read-write. Reset: 1. 1=Enable register sharing among all OHCI and EHCI controllers. 0=Disable register sharing. When register sharing is enabled, values written to the following registers for any enabled OHCI or EHCI controller will be copied to the equivalent register in the other enabled OHCI or EHCI controller(s): <ul style="list-style-type: none"> <li>• D[16,13,12]F0x50</li> <li>• D[16,13,12]F0x80</li> <li>• D[16,13,12]F2x50</li> <li>• D[16,13,12]F2x54</li> </ul>
27	Reserved.
26	<b>FlrEnable: FLR enable.</b> Read-write. Reset: 0. 1=Enable FLR support.
25	<b>OhciL1BlockDisable: OHCI L1 block disable.</b> Read-write. Reset: 0. 1=OHCI L1 block disabled. 0=Enabled.

24	<b>OhciLoopbackControlRegisterEnable: OHCI loopback control register enable.</b> Read-write. Reset: 0. 1=Enable the <a href="#">OHCI[3:1]xF0 [HC Loopback Control]</a> register.												
23:22	<b>OhciPacketBufferThreshold: OHCI packet buffer threshold.</b> Read-write. Reset: 0. BIOS: 11b. Specifies the threshold of OHCI packet buffer. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>48 bytes</td> <td>10b</td> <td>200 bytes</td> </tr> <tr> <td>01b</td> <td>128 bytes</td> <td>11b</td> <td>256 bytes</td> </tr> </tbody> </table>	Bits	Definition	Bits	Definition	00b	48 bytes	10b	200 bytes	01b	128 bytes	11b	256 bytes
Bits	Definition	Bits	Definition										
00b	48 bytes	10b	200 bytes										
01b	128 bytes	11b	256 bytes										
21	Reserved.												
20	<b>OhciAdvancePlistEnable: OHCI advance Plist enable.</b> Read-write. Reset: 0. BIOS: 1. 1=Enable advancing Plist.												
19	<b>SofSyncEn: start of frame sync enable.</b> Read-write. Reset: 0. BIOS: 1. 1=Enable start of frame synchronization for all OHCI controllers.												
18:16	Reserved.												
15	<b>OhciDisconnectDetectionTime: OHCI disconnect detection time.</b> Read-write. Reset: 0. 1=Increase the OHCI disconnect (SE0 state) detection time to 6.16 us. 0=2.33 us.												
14:13	Reserved.												
12	<b>SmiHandshakeDisable: SMI handshake disable.</b> Read-write. Reset: 1. BIOS: 0. 1=The handshake between USB and ACPI is disabled when SMI is requested by USB.												
11:10	<b>OhciPrefetchTimeoutTimer: OHCI prefetch timeout timer.</b> Read-write. Reset: 0. Timeout timer to purge the prefetch data in the data FIFO if the data is not used. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>255 ms</td> <td>10b</td> <td>767 ms</td> </tr> <tr> <td>01b</td> <td>511 ms</td> <td>11b</td> <td>1023 ms</td> </tr> </tbody> </table>	Bits	Definition	Bits	Definition	00b	255 ms	10b	767 ms	01b	511 ms	11b	1023 ms
Bits	Definition	Bits	Definition										
00b	255 ms	10b	767 ms										
01b	511 ms	11b	1023 ms										
9:8	<b>OhciPrefetchCacheLineCount: OHCI prefetch cache line count.</b> Read-write. Reset: 3h. Number of data cache lines prefetch requests for ISO out transaction. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Prefetch is disabled.</td> <td>10b</td> <td>2 cache lines</td> </tr> <tr> <td>01b</td> <td>1 cache line</td> <td>11b</td> <td>3 cache lines</td> </tr> </tbody> </table>	Bits	Definition	Bits	Definition	00b	Prefetch is disabled.	10b	2 cache lines	01b	1 cache line	11b	3 cache lines
Bits	Definition	Bits	Definition										
00b	Prefetch is disabled.	10b	2 cache lines										
01b	1 cache line	11b	3 cache lines										
7	Reserved.												
6	<b>OHCI_CACHE_ENABLE: OHCI cache enable.</b> Read-write. Reset: 1. 1=Enable 64 byte OHCI DMA cache.												
5	<b>HTMSISupportEnable: HT MSI support enable.</b> Read-write. Reset: 0. 1=Enable HyperTransport™ MSI support.												
4	Reserved.												
3	<b>A20StateFunctionEnable: A20 state function enable.</b> Read-write. Reset: 0. 0=Disable A20 Mask. 1=Enable A20 Mask.												
2	Reserved.												
1	Reserved.												
0	<b>OHCI_DYNAMIC_POWER_SAVING_ENABLE: OHCI dynamic power saving enable.</b> Read-write. Reset: 0. 1=Dynamic Power Saving for OHCI is enabled.												

**D[16,13,12]F0x58 Over-Current Control 1**

Bits	Description
------	-------------

31:20	Reserved.																								
19:16	Reserved.																								
15:12	<b>HSPort3OverCurrentControl: HS Port 3 OverCurrent Control.</b> Read-only. Reset: Fh. See: HSPort0OverCurrentControl.																								
11:8	<b>HSPort2OverCurrentControl: HS Port 2 OverCurrent Control.</b> Read-only. Reset: Fh. See: HSPort0OverCurrentControl.																								
7:4	<b>HSPort1OverCurrentControl: HS Port 1 OverCurrent Control.</b> Read-only. Reset: Fh. See: HSPort0OverCurrentControl.																								
3:0	<b>HSPort0OverCurrentControl: HS Port 0 OverCurrent Control.</b> Read-only. Reset: Fh. Specifies the OverCurrent pin mapping for port 0. There are 8 OverCurrent pins (USB_OC[7:0]), any value greater than 7 will disable the OverCurrent function for port 0.																								
	<table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>0000b</td> <td>USB_OC0#</td> <td>0100b</td> <td>USB_OC4#</td> </tr> <tr> <td>0001b</td> <td>USB_OC1#</td> <td>0101b</td> <td>USB_OC5#</td> </tr> <tr> <td>0010b</td> <td>USB_OC2#</td> <td>0110b</td> <td>USB_OC6#</td> </tr> <tr> <td>0011b</td> <td>USB_OC3#</td> <td>0111b</td> <td>USB_OC7#</td> </tr> <tr> <td>1111b-1000b</td> <td>Disable OverCurrent</td> <td></td> <td></td> </tr> </tbody> </table>	Bits	Definition	Bits	Definition	0000b	USB_OC0#	0100b	USB_OC4#	0001b	USB_OC1#	0101b	USB_OC5#	0010b	USB_OC2#	0110b	USB_OC6#	0011b	USB_OC3#	0111b	USB_OC7#	1111b-1000b	Disable OverCurrent		
Bits	Definition	Bits	Definition																						
0000b	USB_OC0#	0100b	USB_OC4#																						
0001b	USB_OC1#	0101b	USB_OC5#																						
0010b	USB_OC2#	0110b	USB_OC6#																						
0011b	USB_OC3#	0111b	USB_OC7#																						
1111b-1000b	Disable OverCurrent																								

**D[16,13,12]F0x68 OHCI Over-Current PME Enable**

Bits	Description
31:5	Reserved.
4:0	<b>OHCIOverCurrentPMEEnable[4:0]: OHCI over-current PME enable.</b> Read-write. Reset: 0. 1=Enables the respective port to be sensitive to over-current conditions as wake-up events when it is owned by OHCI.

**D[16,13,12]F0x74 Target Timeout Control**

Bits	Description
31:24	<b>TimeoutTimer: Timeout Timer.</b> Read-write. Reset: 80h. Timer to control the purge of the delay queue when the master that has initiated the access does not return to complete the transaction. After the timer expires, the queue is invalidated and the next transaction is serviced.
23:8	Reserved.
7:0	<b>RetryCounter: Retry Counter.</b> Read-write. Reset: FFh. Counter to control the purge of the delay queue when the downstream access cycle is not completed within certain time. The transaction is target aborted when the counter expired. 0=The retry counter is disabled.

**D[16,13,12]F0x80 OHCI Spare 1**

Bits	Description
31:16	Reserved.
15:10	Reserved.
9	<b>OhciFsOEFunctional.</b> Read-write. Reset: 0. BIOS: 1. 1=Enable USB 1.1 devices to be functional when EHCI controller is disabled.

8	<b>OHCIArbVldCtl.</b> Read-write. Reset: 0. BIOS: See 2.15.2.5 [OHCI Arbiter Mode]. Specifies that the slot valid signal in OHCI arbiter can be cleared per slot instead of per host. 0=OHCI arbiter can only be cleared per host. 1=OHCI arbiter can be cleared per slot. This bit should be programmed to 1 when (D[16,13,12]F0x80[OHCIArbiterMode]==1xb).																				
7	<b>OHCIGlobalDeepBLinkPowerSavingEnable: OHCI Global Deep B-Link Power Saving Enable.</b> Read-write. Reset: 0. BIOS: 1. 1=OHCI will request to stop the global B-Link clock when the list processor is idle and the cache master is idle. 0=Disable.																				
6	<b>OHCILocalDeepBLinkPowerSavingEnable: OHCI Local Deep B-Link Power Saving Enable.</b> Read-write. Reset: 0. 1=OHCI B-Link clock will be gated when the list processor is idle and the cache master is idle. 0=Disable.																				
5:4	<b>OHCIArbiterMode.</b> Read-write. Reset: 0. BIOS: See 2.15.2.5 [OHCI Arbiter Mode].																				
3:1	<b>OHCIL1EarlyExitTimerSelect: OHCI L1 Early Exit Timer Select.</b> Read-write. Reset: 5h. Timer to control the OHCI L1 early exit. When OHCIL1EarlyExitEnable=1, this field selects the amount of time before SOF that the OHCI controller will request UMI to exit from L1. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>20 us</td> <td>100b</td> <td>60 us</td> </tr> <tr> <td>001b</td> <td>30 us</td> <td>101b</td> <td>70 us</td> </tr> <tr> <td>010b</td> <td>40 us</td> <td>110b</td> <td>80 us</td> </tr> <tr> <td>011b</td> <td>50 us</td> <td>111b</td> <td>90 us</td> </tr> </tbody> </table>	Bits	Definition	Bits	Definition	000b	20 us	100b	60 us	001b	30 us	101b	70 us	010b	40 us	110b	80 us	011b	50 us	111b	90 us
Bits	Definition	Bits	Definition																		
000b	20 us	100b	60 us																		
001b	30 us	101b	70 us																		
010b	40 us	110b	80 us																		
011b	50 us	111b	90 us																		
0	<b>OHCIL1EarlyExitEnable: OHCI L1 Early Exit Enable.</b> Read-write. Reset: 0. BIOS: 1. 1=Enable OHCI L1 Early Exit. 0=Disable.																				

**D[16,13,12]F0xD0 MSI Control**

Bits	Description
31:23	Reserved.
22:20	<b>MSIControl: MSI Control.</b> Read-write. Reset: 0.
19:17	Reserved.
16	<b>MSIControlOut: MSI Control Out.</b> Read-write. Reset: 0. 1=Disable IRQ and use MSI instead.
15:8	<b>NextItemPointer: Next Item Pointer.</b> IF (D[16,13,12]F0x50[FlrEnable]==0 && D[16,13,12]F0x50[HTMSISupportEnable]==1) THEN Value: E4h. ELSIF (D[16,13,12]F0x50[FlrEnable]==1) THEN Value: F0h. ELSE Value: 00h. ENDIF. Pointer to next capability structure.
7:0	<b>MSIUSB: MSI USB.</b> Read-only. Reset: 5h. MSI USB ID.

**D[16,13,12]F0xD4 MSI Address**

Bits	Description
31:2	<b>MsiAddress: MSI Address.</b> Read-write. Reset: 0. System-specified message address.
1:0	Reserved.



**D[16,13,12]F0xD8 MSI Data**

Bits	Description
31:16	Reserved.
15:0	<b>MsiData: MSI Data.</b> Read-write. Reset: 0. System-specified message.

**D[16,13,12]F0xE4 HT MSI Support**

Bits	Description
31:0	<b>HTMSISupport: HT MSI Support.</b> IF (D[16,13,12]F0x50[HTMSISupportEnable]==0) THEN Value: 0h. ELSE Value: A803_0008h. ENDIF. For HyperTransport™ MSI support.

**D[16,13,12]F0xF0 Function Level Reset Capability**

Bits	Description
31:26	Reserved.
25	IF (D[16,13,12]F0x50[FlrEnable]==0) THEN Reserved. ELSE <b>FunctionLevelResetCapability: Function Level Reset Capability.</b> Read-only. Reset: 1. Function level reset support. ENDIF.
24	IF (D[16,13,12]F0x50[FlrEnable]==0) THEN Reserved. ELSE <b>TransactionPendingCapability.</b> Value: 1. 1=Transaction pending feature is supported. ENDIF.
23:16	IF (D[16,13,12]F0x50[FlrEnable]==0) THEN Reserved. ELSE <b>Length.</b> Read-only. Reset: 6h. AF structure length (byte). ENDIF.
15:8	IF (D[16,13,12]F0x50[FlrEnable]==0) THEN Reserved. ELSE <b>NextItemPointer: Next Item Pointer.</b> ENDIF. IF (D[16,13,12]F0x50[HTMSISupportEnable]==0) THEN Value: 00h. ELSE Value: E4h. ENDIF. Pointer to next capability structure. 00h=This is the final item on the list. ENDIF.
7:0	IF (D[16,13,12]F0x50[FlrEnable]==0) THEN Reserved. ELSE <b>CapId.</b> Value: 13h. 13h=Identifies that the function as being AF capable. ENDIF.

**D[16,13,12]F0xF4 Function Level Reset Control**

Bits	Description
31:9	Reserved.
8	IF (D[16,13,12]F0x50[FlrEnable]==0) THEN Reserved. ELSE <b>TransactionPending.</b> Read-only. Reset: 0. 1=Indicates that the Function has issued one or more non-posted transactions which have not been completed, including non-posted transactions that a target has terminated with Retry. 0=Indicates that all non-posted transactions have been completed. ENDIF.
7:1	Reserved.
0	IF (D[16,13,12]F0x50[FlrEnable]==0) THEN Reserved. ELSE <b>InitiateFLR: Initiate FLR.</b> RAZ; Write-1-only. Reset: 0. 1=Initiates Function Level Reset (FLR). ENDIF.

### 3.25.4.1.2 OHCI Memory Mapped IO Registers

See 3.1 [Register Descriptions and Mnemonics] for a description of the register naming convention. The OHCI MMIO base address is specified by D[16,13,12]F0x10[Bar]. For details of OHCI USB controller implementation requirements and behavior consult the *OpenHCI Speciation*.

#### OHCI[3:1]x00 HC Revision

Bits	Description
31:9	Reserved.
8	<b>L: Legacy.</b> Value: 1. 1=Indicates that the legacy support registers are present in this HC.
7:0	<b>REV: Revision.</b> Value:10h. Version of HCI specification.

#### OHCI[3:1]x04 HC Control

Bits	Description												
31:11	Reserved.												
10	<b>RWE: remote wakeup enable.</b> Read-write. Reset: 0. This bit is used by <b>HCD</b> to enable or disable the remote wakeup feature upon the detection of upstream resume signaling. 1=If (OHCI[3:1]x0C[RD]==1), a remote wakeup is signaled to the host system. Setting this bit has no impact on the generation of hardware interrupt.												
9	<b>RWC: remote wakeup connected.</b> Read-write. Reset: 0. This bit indicates whether HC supports remote wakeup signaling. If remote wakeup is supported and used by the system it is the responsibility of system firmware to set this bit during POST. HC clears the bit upon a hardware reset but does not alter it upon a software reset.												
8	<b>IR: interrupt routing.</b> Read-write. Reset: 0. Specifies the routing of interrupts generated by events registered in OHCI[3:1]x0C. 0=All interrupts are routed to the normal host bus interrupt mechanism. 1=Interrupts are routed to the System Management Interrupt.												
7:6	<b>HCFS: host controller functional state.</b> Read-write. Reset: 0. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Bits</th> <th>Definition</th> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>USBRESET</td> <td>10b</td> <td>USBOPERATIONAL</td> </tr> <tr> <td>01b</td> <td>USBRESUME</td> <td>11b</td> <td>USBSUSPEND</td> </tr> </tbody> </table> A transition to USBOPERATIONAL from another state causes SOF generation to begin 1 ms later.	Bits	Definition	Bits	Definition	00b	USBRESET	10b	USBOPERATIONAL	01b	USBRESUME	11b	USBSUSPEND
Bits	Definition	Bits	Definition										
00b	USBRESET	10b	USBOPERATIONAL										
01b	USBRESUME	11b	USBSUSPEND										
5	<b>BLE: bulk list enable.</b> Read-write. Reset: 0. 1=Enable the processing of the Bulk list in the next frame.												
4	<b>CLE: control list enable.</b> Read-write. Reset: 0. 1=Enable the processing of the Control list in the next Frame.												
3	<b>IE: isochronous enable.</b> Read-write. Reset: 0. This bit is used by <b>HCD</b> to enable/disable processing of isochronous EDs. 1=The HC continues processing the EDs when it finds an Isochronous ED. 0=HC halts processing of the periodic list (which now contains only isochronous EDs) and begins processing the Bulk/Control lists. Setting this bit takes effect in the next Frame.												
2	<b>PLE: periodic list enable.</b> Read-write. Reset: 0. 1=Enable the processing of the periodic list in the next Frame.												

1:0	<b>CBSR: control bulk service ratio.</b> Read-write. Reset: 0. This specifies the service ratio between Control and Bulk EDs.			
	<u>Bits</u>	<u>Service Ratio</u>	<u>Bits</u>	<u>Service Ratio</u>
	00b	1:1	10b	3:1
	01b	2:1	11b	4:1

**OHCI[3:1]x08 HC Command Status**

Bits	Description
31:18	Reserved.
17:16	<b>SOC: SchedulingOverrunCount.</b> Read-write. Reset: 0. These bits increment on each scheduling overrun error.
15:4	Reserved.
3	<b>OCR: OwnershipChange.</b> Read-write. Reset: 0. When set, HC will set the <a href="#">OHCI[3:1]x0C[OC]</a> .
2	<b>BLF: BulkListFilled.</b> Read-write. Reset: 0. This bit is used to indicate whether there are any TDs on the Bulk list.
1	<b>CLF: ControlListFilled.</b> Read-write. Reset: 0. This bit is used to indicate whether there are any TDs on the Control list.
0	<b>HCR: HostControllerReset.</b> Read; write-1-only; cleared-when-done. Reset: 0. This bit is set by <a href="#">HCD</a> to initiate a software reset of HC.

**OHCI[3:1]x0C HC Interrupt Status**

Bits	Description
31	Reserved.
30	<b>OC: OwnershipChange.</b> Read-write. Reset: 0. 1= The HC detected that the <a href="#">HCD</a> set <a href="#">OHCI[3:1]x08[OCR]</a> .
29:7	Reserved.
6	<b>RHSC: RootHubStatusChange.</b> Read-write. Reset: 0. This bit is set when the content of <a href="#">OHCI[3:1]x50</a> or the content of any of <a href="#">OHCI[3:1]x[60:54:step4]</a> has changed.
5	<b>FNO: FrameNumberOverflow.</b> Read-write. Reset: 0. This bit is set when <a href="#">OHCI[3:1]x3C[FN[15]]</a> changes value.
4	<b>UE: UnrecoverableError.</b> Read-write. Reset: 0. 1=The HC detected a system error not related to USB. HC should not proceed with any processing or signaling before the system error has been corrected. <a href="#">HCD</a> clears this bit after HC has been reset.
3	<b>RD: ResumeDetected.</b> Read-write. Reset: 0. 1=The HC detected that a device on the USB is asserting resume signaling. It is the transition from no resume signaling to resume signaling causing this bit to be set. This bit is not set when <a href="#">HCD</a> sets the USBRESUME state.
2	<b>SF: StartOfFrame.</b> Read-write. Reset: 0. This bit is set by HC at each start of a frame and after the update of HccaFrameNumber. HC also generates a SOF token at the same time.
1	<b>WDH: WritebackDoneHead.</b> Read-write. Reset: 0. This bit is set immediately after HC has written HcDoneHead to HccaDoneHead. Further updates of the HccaDoneHead will not occur until this bit has been cleared. <a href="#">HCD</a> should only clear this bit after it has saved the content of HccaDoneHead.

0	<b>SO: SchedulingOverrun.</b> Read-write. Reset: 0. This bit is set when the USB schedule for the current Frame overruns and after the update of HccaFrameNumber. A scheduling overrun will also cause the SchedulingOverrunCount of HcCommandStatus to be incremented.
---	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### **OHCI[3:1]x10 HC Interrupt Enable**

Bits	Description
31	<b>MIE: MasterInterruptEnable.</b> Read; write-1-only. Reset: 0. 1=Enables interrupt generation due to events specified in the other bits of this register. See corresponding field in <a href="#">OHCI[3:1]x14</a> for disable.
30	<b>OC: OwnershipChange.</b> Read; write-1-only. Reset: 0. 1=Enable interrupt generation due to Ownership Change. See corresponding field in <a href="#">OHCI[3:1]x14</a> for disable.
29:7	Reserved.
6	<b>RHSC: RootHubStatusChange.</b> Read; write-1-only. Reset: 0. 1=Enable interrupt generation due to Root Hub Status Change. See corresponding field in <a href="#">OHCI[3:1]x14</a> for disable.
5	<b>FNO: FrameNumberOverflow.</b> Read; write-1-only. Reset: 0. 1=Enable interrupt generation due to Frame Number Overflow. See corresponding field in <a href="#">OHCI[3:1]x14</a> for disable.
4	<b>UE: UnrecoverableError.</b> Read; write-1-only. Reset: 0. 1=Enable interrupt generation due to Unrecoverable Error. See corresponding field in <a href="#">OHCI[3:1]x14</a> for disable.
3	<b>RD: ResumeDetect.</b> Read; write-1-only. Reset: 0. 1=Enable interrupt generation due to Resume Detect. See corresponding field in <a href="#">OHCI[3:1]x14</a> for disable.
2	<b>SF: StartofFrame.</b> Read; write-1-only. Reset: 0. 1=Enable interrupt generation due to Start of Frame. See corresponding field in <a href="#">OHCI[3:1]x14</a> for disable.
1	<b>WDH: HcDoneHeadWriteback.</b> Read; write-1-only. Reset: 0. 1=Enable interrupt generation due to HcDoneHead Writeback. See corresponding field in <a href="#">OHCI[3:1]x14</a> for disable.
0	<b>SO: SchedulingOverrun.</b> Read; write-1-only. Reset: 0. 1=Enable interrupt generation due to Scheduling Overrun. See corresponding field in <a href="#">OHCI[3:1]x14</a> for disable.

### **OHCI[3:1]x14 HC Interrupt Disable**

Reset: 8000\_0000h.

Bits	Description
31	<b>MIE: MasterInterruptEnable.</b> Read; write-1-only. 1=Disables interrupt generation due to events specified in the other bits of this register. See corresponding field in <a href="#">OHCI[3:1]x10</a> for enable.
30	<b>OC: OwnershipChange.</b> Read; write-1-only. 1=Disable interrupt generation due to Ownership Change. See corresponding field in <a href="#">OHCI[3:1]x10</a> for enable.
29:7	Reserved.
6	<b>RHSC: RootHubStatusChange.</b> Read; write-1-only. 1=Disable interrupt generation due to Root Hub Status Change. See corresponding field in <a href="#">OHCI[3:1]x10</a> for enable.
5	<b>FNO: FrameNumberOverflow.</b> Read; write-1-only. 1=Disable interrupt generation due to Frame Number Overflow. See corresponding field in <a href="#">OHCI[3:1]x10</a> for enable.
4	<b>UE: UnrecoverableError.</b> Read; write-1-only. 1=Disable interrupt generation due to Unrecoverable Error. See corresponding field in <a href="#">OHCI[3:1]x10</a> for enable.

3	<b>RD: ResumeDetect.</b> Read; write-1-only. 1=Disable interrupt generation due to Resume Detect. See corresponding field in <a href="#">OHCI[3:1]x10</a> for enable.
2	<b>SF: StartoffFrame.</b> Read; write-1-only. 1=Disable interrupt generation due to Start of Frame. See corresponding field in <a href="#">OHCI[3:1]x10</a> for enable.
1	<b>WDH: HcDoneHeadWriteback.</b> Read; write-1-only. 1=Disable interrupt generation due to HcDone-Head Writeback. See corresponding field in <a href="#">OHCI[3:1]x10</a> for enable.
0	<b>SO: SchedulingOverrun.</b> Read; write-1-only. 1=Disable interrupt generation due to Scheduling Overrun. See corresponding field in <a href="#">OHCI[3:1]x10</a> for enable.

### **OHCI[3:1]x18 HC HCCA**

Bits	Description
31:8	<b>HCCA: HostControllerCommunicationArea.</b> Read-write. Reset: 0. This is the base address of the Host Controller Communication Area
7:0	Reserved.

### **OHCI[3:1]x1C HC Period Current ED**

Bits	Description
31:4	<b>PCED: PeriodCurrentED.</b> Read-write. Reset: 0. This is used by HC to point to the head of one of the Periodic lists which will be processed in the current Frame. The content of this register is updated by HC after a periodic ED has been processed. <a href="#">HCD</a> may read the content in determining which ED is currently being processed at the time of reading.
3:0	Reserved.

### **OHCI[3:1]x20 HC Control Head ED**

Bits	Description
31:4	<b>CHED: ControlHeadED.</b> Read-write. Reset: 0. HC traverses the Control list starting with the HcControlHeadED pointer. The content is loaded from HCCA during the initialization of HC.
3:0	Reserved.

### **OHCI[3:1]x24 HC Control Current ED**

Bits	Description
31:4	<b>CCED: ControlCurrentED.</b> Read-write. Reset: 0. This pointer is advanced to the next ED after serving the present one.
3:0	Reserved.

### **OHCI[3:1]x28 HC Bulk Head ED**

Bits	Description
------	-------------

31:4	<b>BHED: BulkHeadED.</b> Read-write. Reset: 0. HC traverses the Bulk list starting with the HcBulk-HeadED pointer. The content is loaded from HCCA during the initialization of HC.
3:0	Reserved.

**OHCI[3:1]x2C HC BulkCurrent ED**

Bits	Description
31:4	<b>BCED: BulkCurrentED.</b> Read-write. Reset: 0. This is advanced to the next ED after the HC has served the present one.
3:0	Reserved.

**OHCI[3:1]x30 HC Done Head**

Bits	Description
31:4	<b>DH: DoneHead.</b> Read-write. Reset: 0. When a TD is completed, HC writes the content of HcDone-Head to the NextTD field of the TD. HC then overwrites the content of HcDoneHead with the address of this TD. This is set to 0 whenever HC writes the content of this register to HCCA. It also sets <a href="#">OHCI[3:1]x0C[WDH]</a> .
3:0	Reserved.

**OHCI[3:1]x34 HC Frame Interval**

Bits	Description
31	<b>FIT: frame interval toggle.</b> Read-write. Reset: 0. <a href="#">HCD</a> toggles this bit whenever it loads a new value to <a href="#">OHCI[3:1]x34[FI]</a> .
30:16	<b>FSMPS: FS largest data packet.</b> Read-write. Reset: 0. This field specifies a value which is loaded into the Largest Data Packet Counter at the beginning of each frame. The counter value represents the largest amount of data in bits which can be sent or received by the HC in a single transaction at any given time without causing scheduling overrun. The field value is calculated by the <a href="#">HCD</a> .
15:14	Reserved.
13:0	<b>FI: frame interval.</b> Read-write; updated-by-hardware. Reset: 2EDFh. This specifies the interval between two consecutive SOFs in bit times. <a href="#">HCD</a> should store the current value of this field before resetting HC. Programming <a href="#">OHCI[3:1]x08[HCR]=1</a> causes the HC to overwrite this field with the reset value. <a href="#">HCD</a> may choose to restore the stored value upon the completion of the reset sequence.

**OHCI[3:1]x38 HC Frame Remaining**

Bits	Description
31	<b>FRT: FrameRemainingToggle.</b> Read-only. Reset: 0. This bit is loaded from the FrameIntervalToggle field of HcFmInterval whenever FrameRemaining reaches 0. This bit is used by <a href="#">HCD</a> for the synchronization between FrameInterval and FrameRemaining.
30:14	Reserved.

13:0	<b>FR: FrameRemaining.</b> Read-only. Reset: 0. This counter is decremented at each bit time. When it reaches 0, it is reset by loading the FrameInterval value specified in HcFmInterval at the next bit time boundary. When entering the USBOPERATIONAL state, HC re-loads the content with the FrameInterval field of HcFmInterval and uses the updated value from the next SOF.
------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**OHCI[3:1]x3C HC Frame Number**

Bits	Description
31:16	Reserved.
15:0	<b>FN[15:0]: FrameNumber.</b> Read-only. Reset: 0. This is incremented when HcFmRemaining is re-loaded. It will be rolled over to 0h after FFFFh. When entering the USBOPERATIONAL state, this will be incremented automatically. The content will be written to HCCA after HC has incremented FrameNumber at each frame boundary and sent a SOF but before HC reads the first ED in that Frame. After writing to HCCA, HC will set OHCI[3:1]x0C[SF].

**OHCI[3:1]x40 HC Periodic Start**

Bits	Description
31:14	Reserved.
13:0	<b>PS: PeriodicStart.</b> Read-write. Reset: 0. This bit is set by HCD during the HC initialization. The value is calculated roughly as 10% off from HcFmInterval. A typical value will be 3E67h. When HcFmRemaining reaches the value specified, processing of the periodic lists will have priority over Control/Bulk processing. HC will therefore start processing the interrupt list after completing the current Control or Bulk transaction that is in progress.

**OHCI[3:1]x44 HC LS Threshold**

Bits	Description
31:12	Reserved.
11:0	<b>LST: LSThreshold.</b> Read-write. Reset: 628h. This field contains a value which is compared to the FrameRemaining field prior to initiating a Low Speed transaction. The transaction is started only if FrameRemaining is equal or greater than this field. The value is calculated by HCD with the consideration of transmission and setup overhead.

**OHCI[3:1]x48 HC RH Descriptor A**

**Table 233:** OHCI[3:1]x48 reset values

Register	Reset
OHCI1x48	0200_0B04h
OHCI2x48	0200_0B04h
OHCI3x48	0200_0B02h

Bits	Description
------	-------------

31:24	<b>POTPGT: PowerOnToPowerGoodTime.</b> Read-write. This field specifies the duration <b>HCD</b> has to wait before accessing a powered-on port of the Root Hub. It is implementation-specific. The unit of time is 2 ms. The duration is calculated as POTPGT * 2 ms.
23:13	Reserved.
12	<b>NOCP: NoOverCurrentProtection.</b> Read-write. Specifies how the over-current status for the Root Hub ports is reported. When this bit is cleared, the OverCurrentProtectionMode field specifies global or per-port reporting. 0=Over-current status is reported collectively for all downstream ports. 1=No over-current protection supported.
11	<b>OCPM: OverCurrentProtectionMode.</b> Read-write. Specifies how the over-current status for the Root Hub ports are reported. This field is valid only if (OHCI[3:1]x48[NOCP]==0). 0=Over-current status is reported collectively for all downstream ports. 1=Over-current status is reported on a per-port basis.
10	<b>DT: DeviceType.</b> Value: 0. 0=Specifies that the Root Hub is not a compound device.
9	<b>NPS: NoPowerSwitching.</b> Read-write. Reset: 1. Specifies whether power switching is supported or ports are always powered. 0=Ports are power switched. 1=Ports are always powered on when the HC is powered on.
8	<b>PSM: PowerSwitchingMode.</b> Read-write. Reset: 1. Specifies how the power switching of the Root Hub ports is controlled. 0=All ports are powered at the same time. 1=Each port is powered individually.
7:0	<b>NDP: NumberDownstreamPorts.</b> Read-only. Specifies the number of downstream ports supported by the Root Hub.

### OHCI[3:1]x4C HC RH Descriptor B

Bits	Description												
31:16	<p><b>PPCM: PortPowerControlMask.</b> Read-write. Reset: 0. Each bit indicates if a port is affected by a global power control command when PSM is set. 1=The port's power state is only affected by per-port power control (SetPortPower/ClearPortPower). 0=The port is controlled by the global power switch (SetrGlobalPower/ClearGlobalPower). If (OHCI[3:1]x48[PSM]==0), this field is not valid.</p> <table border="1"> <thead> <tr> <th>Bit</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>[0]</td> <td>Reserved</td> </tr> <tr> <td>[1]</td> <td>Ganged-power mask on Port #1</td> </tr> <tr> <td>[2]</td> <td>Ganged-power mask on Port #2</td> </tr> <tr> <td>[14:3]</td> <td>...</td> </tr> <tr> <td>[15]</td> <td>Ganged-power mask on Port #15</td> </tr> </tbody> </table>	Bit	Definition	[0]	Reserved	[1]	Ganged-power mask on Port #1	[2]	Ganged-power mask on Port #2	[14:3]	...	[15]	Ganged-power mask on Port #15
Bit	Definition												
[0]	Reserved												
[1]	Ganged-power mask on Port #1												
[2]	Ganged-power mask on Port #2												
[14:3]	...												
[15]	Ganged-power mask on Port #15												
15:0	<p><b>DR: DeviceRemovable.</b> Read-write. Reset: 0. Each bit is dedicated to a port of the Root Hub. 0=The attached device is removable. 1=The attached device is not removable.</p> <table border="1"> <thead> <tr> <th>Bit</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>[0]</td> <td>Reserved</td> </tr> <tr> <td>[1]</td> <td>Device attached to Port #1</td> </tr> <tr> <td>[2]</td> <td>Device attached to Port #2</td> </tr> <tr> <td>[14:3]</td> <td>...</td> </tr> <tr> <td>[15]</td> <td>Device attached to Port #15</td> </tr> </tbody> </table>	Bit	Definition	[0]	Reserved	[1]	Device attached to Port #1	[2]	Device attached to Port #2	[14:3]	...	[15]	Device attached to Port #15
Bit	Definition												
[0]	Reserved												
[1]	Device attached to Port #1												
[2]	Device attached to Port #2												
[14:3]	...												
[15]	Device attached to Port #15												



**OHCI[3:1]x50 HC RH Status**

Bits	Description
31	<b>CRWE: ClearRemoteWakeupEnable.</b> RAZ; write-1-only. Reset: 0. Writing a 1 clears DeviceRemoveWakeupEnable.
30:18	Reserved.
17	<b>OCIC: OverCurrentIndicatorChange.</b> Read; write-1-to-clear. Reset: 0. 1=A change of state occurred in OHCI[3:1]x50[OCI].
16	<b>LPSC: LocalPowerStatusChange.</b> RAZ; write. Reset: 0. IF (READ) THEN LocalPowerStatusChange. The Root Hub does not support the local power status feature; this bit is always read as 0. ELSE SetGlobalPower. In global power mode (OHCI[3:1]x48[PSM]==0), this bit is written to 1 to turn on power to all ports (clear PortPowerStatus). In per-port power mode (OHCI[3:1]x48[PSM]==1), it sets PortPowerStatus only on ports whose (OHCI[3:1]x4C[PPCM]==0). ENDIF.
15	<b>DRWE: DeviceRemoteWakeupEnable.</b> Read-write. Reset: 0. IF (READ) THEN DeviceRemoteWakeupEnable. This bit enables the ConnectStatusChange bit as a resume event, causing a USBsuspend to USBResume state transition and setting the ResumeDetected interrupt. 0=ConnectStatusChange is not a remote wakeup event. 1=ConnectStatusChange is a remote wakeup event. ELSE SetRemoteWakeupEnable. 1=Set DeviceRemoveWakeupEnable. 0=No effect. ENDIF.
14:2	Reserved.
1	<b>OCI: OverCurrentIndicator.</b> Read-only. Reset: 0. This bit reports over-current conditions when the global reporting is implemented. 1=An over-current condition exists. 0=All power operations are normal. If per-port over-current protection is implemented, this bit is always 0
0	<b>LPS: LocalPowerStatus.</b> RAZ; write. Reset: 0. IF (READ) THEN LocalPowerStatus. The Root Hub does not support the local power status feature and this bit is always read as 0. ELSE ClearGlobalPower. In global power mode (OHCI[3:1]x48[PSM]==0), writing 1 to this bit turns off power to all ports (clear PortPowerStatus). In per-port power mode (OHCI[3:1]x48[PSM]==1), it clears PortPowerStatus only on ports whose (OHCI[3:1]x4C[PPCM]==0). ENDIF.

**OHCI[3:1]x[60:54:step4] HC RH Port Status [4:1]**

Bits	Description
31:21	Reserved.
20	<b>PRSC: PortResetStatusChange.</b> Read; write-1-to-clear. Reset: 0. This bit is set at the end of the 10 ms port reset signal. 0=Port reset is not complete. 1=Port reset is complete.
19	<b>OCIC: PortOverCurrentIndicatorChange.</b> Read; write-1-to-clear. Reset: 0. This bit is valid only if over-current conditions are reported on a per-port basis. This bit is set when Root Hub changes the PortOverCurrentIndicator bit. 0=No change in PortOverCurrentIndicator. 1=PortOverCurrentIndicator has changed.

18	<b>PSSC: PortSuspendStatusChange.</b> Read; write-1-to-clear. Reset: 0. This bit is set when the full resume sequence has been completed. This sequence includes the 20 s resume pulse, LS EOP, and 3 ms resynchronization delay. This bit is also cleared when ResetStatusChange is set. 0=Resume is not completed. 1=Resume completed.
17	<b>PESC: PortEnableStatusChange.</b> Read; write-1-to-clear. Reset: 0. This bit is set when hardware events cause the PortEnableStatus bit to be cleared. Changes from HCD writes do not set this bit. 0=No change in PortEnableStatus. 1=Change in PortEnableStatus.
16	<b>CSC: ConnectStatusChange.</b> Read; write-1-to-clear. Reset: 0. This bit is set whenever a connect or disconnect event occurs. If CurrentConnectStatus is cleared when a SetPortReset, SetPortEnable, or SetPortSuspend write occurs, this bit is set to force the driver to re-evaluate the connection status since these writes should not occur if the port is disconnected. 0=No change in CurrentConnectStatus. 1=Change in CurrentConnectStatus. Note: If the OHCI[3:1]x4C[DR] bit is set, this bit is set only after a Root Hub reset to inform the system that the device is attached.
15:10	Reserved.
9	<b>LSDA.</b> Read; write-1-only. Reset: X. IF (READ) THEN LowSpeedDeviceAttached. This field is valid only when the CurrentConnectStatus is set. This bit indicates the speed of the device attached to this port. 1=A Low Speed device is attached to this port. 0=A Full Speed device is attached to this port. ELSE ClearPortPower. Writing 1 to this bit clears the PortPowerStatus bit. Writing 0 has no effect. ENDIF.
8	<b>PPS.</b> Read; write-1-only. Reset: 0. IF (READ) THEN PortPowerStatus. This bit reflects the port's power status, regardless of the type of power switching implemented. This bit is cleared if an over-current condition is detected. HCD sets this bit by writing SetPortPower or SetGlobalPower. HCD clears this bit by writing ClearPortPower or ClearGlobalPower. Which power control switches are enabled is determined by PSM and OHCI[3:1]x4C[PPCM]. In global switching mode, (PSM=0), only SetGlobalPower/ClearGlobalPower controls this bit. In per-port power switching (PSM=1), if the OHCI[3:1]x4C[PPCM] bit for the port is set, only SetPortPower/ClearPortPower commands are enabled. If the mask is not set, only SetGlobalPower/ClearGlobalPower commands are enabled. When port power is disabled, CurrentConnectStatus, PortEnableStatus, PortSuspendStatus, and PortResetStatus should be reset. 0=Port power is off. 1=Port power is on. ELSE SetPortPower. Writing 1 to this bit sets the PortPowerStatus bit. Writing 0 has no effect. ENDIF. Note: This bit always reads 1b if power switching is not supported.
7:5	Reserved.
4	<b>PRS.</b> Read; write-1-only. Reset: 0. IF (READ) THEN PortResetStatus. When this bit is set by a write to SetPortReset, port reset signaling is asserted. When reset is completed, this bit is cleared when PortResetStatusChange is set. This bit cannot be set if CurrentConnectStatus is cleared. 0=Port reset signal is not active. 1=Port reset signal is active. ELSE SetPortReset. The HCD sets the port reset signaling by writing a 1 to this bit. Writing a 0 has no effect. If CurrentConnectStatus is cleared, this write does not set PortResetStatus, but instead sets ConnectStatusChange. This informs the driver that it attempted to reset a disconnected port. ENDIF.

3	<p><b>POCI.</b> Read-write. Reset: 0.  IF (READ) THEN PortOverCurrentIndicator. This bit is only valid when the Root Hub is configured in such a way that over-current conditions are reported on a per-port basis. If per-port over-current reporting is not supported, this bit is set to 0. If cleared, all power operations are normal for this port. If set, an over-current condition exists on this port. This bit always reflects the over-current input signal. 0=No over-current condition. 1=Over-current condition detected.  ELSE ClearSuspendStatus. The <b>HCD</b> writes a 1 to initiate a resume. Writing a 0 has no effect. A resume is initiated only if PortSuspendStatus is set.  ENDIF.</p>
2	<p><b>PSS.</b> Read-write. Reset: 0.  IF (READ) THEN PortSuspendStatus. This bit indicates the port is suspended or in the resume sequence. It is set by a SetSuspendState write and cleared when PortSuspendStatusChange is set at the end of the resume interval. This bit cannot be set if CurrentConnectStatus is cleared. This bit is also cleared when PortResetStatusChange is set at the end of the port reset or when the HC is placed in the USBRESUME state. If an upstream resume is in progress, it should propagate to the HC. 0=Port is not suspended. 1=Port is suspended.  ELSE SetPortSuspend. The <b>HCD</b> sets the PortSuspendStatus bit by writing a 1 to this bit. Writing a 0 has no effect. If CurrentConnectStatus is cleared, this write does not set PortSuspendStatus; instead it sets ConnectStatusChange. This informs the driver that it attempted to suspend a disconnected port.  ENDIF.</p>
1	<p><b>PES.</b> Read-write. Reset: 0.  IF (READ) THEN PortEnableStatus. This bit indicates whether the port is enabled or disabled. The Root Hub may clear this bit when an over-current condition, disconnect event, switched-off power, or operational bus error such as babble is detected. This change also causes PortEnabledStatusChange to be set. <b>HCD</b> sets this bit by writing SetPortEnable and clears it by writing ClearPortEnable. This bit cannot be set when CurrentConnectStatus is cleared. This bit is also set, if not already, at the completion of a port reset when ResetStatusChange is set or port suspend when SuspendStatusChange is set. 0=Port is disabled. 1=Port is enabled.  ELSE SetPortEnable. The <b>HCD</b> sets PortEnableStatus by writing a 1. Writing a 0 has no effect. If CurrentConnectStatus is cleared, this write does not set PortEnableStatus, but instead sets ConnectStatusChange. This informs the driver that it attempted to enable a disconnected port.  ENDIF.</p>
0	<p><b>CCS.</b> Read-write. Reset: 0.  IF (READ) THEN CurrentConnectStatus. This bit reflects the current state of the downstream port. 0=No device connected. 1=Device connected.  ELSE ClearPortEnable. The <b>HCD</b> writes a 1 to this bit to clear the PortEnableStatus bit. Writing a 0 has no effect. CurrentConnectStatus is not affected by any write.  ENDIF.  Note: This bit is always read 1b when the attached device is nonremovable (<b>OHCI[3:1]x4C[DR]</b>).</p>

### OHCI[3:1]xF0 HC Loopback Control

Reset: 0000\_0000h.

IF (**D[16,13,12]F0x50**[OhciLoopbackControlRegisterEnable]==1) THEN then this register is defined; otherwise it is reserved. ENDIF.

Bits	Description
31	<b>LoopBackTestDone.</b> Read-only. 1=Loop back is done. Cleared when software clears LoopBackTestStart.

30:16	<b>LoopBackTestStatus.</b> Read-only. Specifies the loop back test status for the port under test. For each bit, 0=Fail. 1=Pass. Software can only check these status bits when LoopBackTestDone is set by the host controller. These bits can only be cleared when software clears LoopBackTestStart. <table border="1"> <thead> <tr> <th><u>Bit</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>[0]</td> <td>Port 0</td> </tr> <tr> <td>[1]</td> <td>Port 1</td> </tr> <tr> <td>...</td> <td>...</td> </tr> <tr> <td>[14]</td> <td>Port 14</td> </tr> </tbody> </table>	<u>Bit</u>	<u>Definition</u>	[0]	Port 0	[1]	Port 1	...	...	[14]	Port 14
<u>Bit</u>	<u>Definition</u>										
[0]	Port 0										
[1]	Port 1										
...	...										
[14]	Port 14										
15:8	<b>LoopBackTestData.</b> Read-write. 1-byte test data pattern for transmit and receive logic check the received data to match with this data pattern.										
7:4	<b>PortUnderTest: Port Under Test.</b> Read-write. Software selects the port under test through these bits. Software should only program the port number to the range of ports that OHCI supports. <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Port 0</td> </tr> <tr> <td>1h</td> <td>Port 1</td> </tr> <tr> <td>...</td> <td>...</td> </tr> <tr> <td>Fh</td> <td>Port 15</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	0h	Port 0	1h	Port 1	...	...	Fh	Port 15
<u>Bits</u>	<u>Definition</u>										
0h	Port 0										
1h	Port 1										
...	...										
Fh	Port 15										
3:1	Reserved.										
0	<b>LoopBackTestStart.</b> Read-write. 1=Start the loop back test. 0=Clear out all the test status before starting the next loop.										

**OHCI[3:1]x100 HCE Control**

Reset: 0000\_0000h.

Bits	Description
31:9	Reserved.
8	<b>A20State.</b> Read-write. Indicates current state of Gate A20 on keyboard controller. Used to compare against value written to 60h when GateA20Sequence is active.
7	<b>IRQ12Active.</b> Read; write-1-to-clear. 1=A positive transition on IRQ12 from the keyboard controller has occurred.
6	<b>IRQ1Active.</b> Read; write-1-to-clear. 1=A positive transition on IRQ1 from the keyboard controller has occurred.
5	<b>GateA20Sequence.</b> Read-write. Set by HC when (IO port 64h==D1h). Cleared by HC on write to (IO port 64h!=D1h).
4	<b>ExternalIRQEn.</b> Read-write. 1=Enable IRQ1 and IRQ12 from the keyboard controller to cause an emulation interrupt. The function controlled by this bit is independent of the setting of the Emulation-Enable bit in this register.
3	<b>IRQEn.</b> Read-write. 1=Enable the HC generation of IRQ1 or IRQ12 as long as the (OHCI[3:1]x10C[OutputFull] ==1). IF (OHCI[3:1]x10C[AuxOutputFull]==0) THEN IRQ1 is generated. ELSE IRQ12 is generated.
2	<b>CharacterPending.</b> Read-write. 1=An emulation interrupt is generated when the OutputFull bit of the HceStatus register is set to 0.
1	<b>EmulationInterrupt.</b> Read-only. This bit is a static decode of the emulation interrupt condition.

0	<b>EmulationEnable.</b> Read-write. 1=Enable HC legacy keyboard and mouse emulation. The HC decodes accesses to IO registers 60h and 64h and generates IRQ1 and/or IRQ12 when appropriate. Additionally, the HC generate s an emulation interrupt at appropriate times to invoke the emulation software.
---	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**OHCI[3:1]x104 HCE Input**

Bits	Description
31:8	Reserved.
7:0	<b>InputData.</b> Read-write. Reset: 0. This register holds data that is written to IO ports 60h and 64h.

**OHCI[3:1]x108 HCE Output**

Bits	Description
31:8	Reserved.
7:0	<b>OutputData.</b> Read-write. Reset: 0. This register hosts data that is returned when an IO read of port 60h is performed by application software.

**OHCI[3:1]x10C HCE Status**

Reset: 0000\_0000h.

Bits	Description
31:8	Reserved.
7	<b>Parity.</b> Read-write. 1=Indicates parity error on keyboard/mouse data.
6	<b>TimeOut.</b> Read-write. 1=Indicates a timeout.
5	<b>AuxOutputFull.</b> Read-write. IRQ12 is asserted whenever this bit is set to 1 and OutputFull is set to 1 and the IRQEn bit is set.
4	<b>InhibitSwitch.</b> Read-write. This bit reflects the state of the keyboard inhibit switch and is set if the keyboard is NOT inhibited.
3	<b>CmdData.</b> Read-write. The HC sets this bit to 0 on an IO write to port 60h and to 1 on an IO write to port 64h.
2	<b>Flag.</b> Read-write. Nominally used as a system flag by software to indicate a warm or cold boot.
1	<b>InputFull.</b> Read-write. Except for the case of a Gate A20 sequence, this bit is set to 1 on an IO write to address 60h or 64h. While this bit is set to 1 and emulation is enabled, an emulation interrupt condition exists.
0	<b>OutputFull.</b> Read-write. The HC sets this bit to 0 on a read of IO port 60h. If IRQEn is set and AuxOutputFull is set to 0, then an IRQ1 is generated as long as this bit is set to 1. If IRQEn is set and AuxOutputFull is set to 1, then an IRQ12 is generated as long as this bit is set to 1. While this bit is 0 and CharacterPending in HceControl is set to 1, an emulation interrupt condition exists.

**3.25.4.2 USB 2.0 (EHCI)**The EHCI controllers can be disabled by programming [PMxEF](#).

**3.25.4.2.1 Devices 16h, 13h, 12h Function 2 (EHCI) 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[16,13,12]F2x00 Device/Vendor ID**

Bits	Description
31:16	<b>DeviceID: Device ID.</b> Read-only. Reset: 7808h. This 16-bit field is assigned by the device manufacturer and identifies the type of device.
15:0	<b>VendorID: Vendor ID.</b> Read-only. Reset: 1022h.

**D[16,13,12]F2x04 Status/Command**

Bits	Description
31	<b>DetectedParityError: Detected Parity Error.</b> Read; write-1-to-clear. Reset: 0. 1=The EHCI controller detects a parity error.
30	<b>SignaledSystemError: Signaled System Error.</b> Read; write-1-to-clear. Reset: 0. 1=The EHCI controller asserted SERR#.
29	<b>ReceivedMasterAbort: Received Master Abort.</b> Read; write-1-to-clear. Reset: 0. 1=The EHCI controller received a PCI master abort while acting as a PCI master.
28	<b>ReceivedTargetAbort: Received Target Abort.</b> Read; write-1-to-clear. Reset: 0. 1=The EHCI controller generated PCI cycle (EHCI controller is the PCI master) is aborted by a PCI target.
27	<b>SignaledTargetAbort: Signaled Target Abort.</b> Read; write-1-to-clear. Read. Reset: 0. 1=The EHCI controller signals Target Abort.
26:25	<b>DEVSELtiming: DEVSEL timing.</b> Value: 01b. 01b=Medium timing.
24	<b>MasterDataParityError: Master Data Parity Error.</b> Read; write-1-to-clear. Reset: 0. 1=EHCI controller detects PERR# being asserted while acting as PCI master, regardless whether PERR# was driven by EHCI controller or not.
23	<b>FastBack2BackCapable: Fast Back-to-Back Capable.</b> Value: 1. 1=Indicates Fast Back-to-Back capable.
22	Reserved.
21	<b>Capable66MHz: 66 MHz Capable.</b> Value: 1. 1=Indicates 66 MHz capable.
20	<b>CapabilitiesList: Capabilities List.</b> Read-only. Reset: 1. 1=Indicates that the value read at D[16,13,12]F2x34 [Capability Pointer] is a pointer in Configuration Space to a linked list of new capabilities.
19	<b>InterruptStatus: Interrupt Status.</b> Read-only. Reset: 0. This bit reflects the state of the interrupt in the device/function.
18:11	Reserved.
10	<b>InterruptDisable: Interrupt Disable.</b> Read-write. Reset: 0. 0=Enable the assertion of the device/function's INTx# signal. 1=Disable the assertion of the device/function's INTx# signal.
9	<b>FastBack2BackEnable: Fast Back-to-Back Enable.</b> Read-write. Reset: 0. 0=Only fast back-to-back transactions to the same agent are allowed. 1=The master is allowed to generate fast back-to-back transactions to different agents.

8	<b>SERREnable: SERR# Enable.</b> Read-write. Reset: 0. 0=Disable the SERR# driver. 1=Enable the SERR# driver.
7	RAZ.
6	<b>ParityEnable: Parity Enable.</b> Read-write. Reset: 0. 0=The device sets its DetectedParityError bit when an error is detected, but continues normal operations without asserting PERR#. 1=The device must take its normal action when a parity error is detected.
5	<b>VgaPaletteRegisterAccesses: VGA palette register accesses.</b> Value: 0. 0=The device treats palette write accesses the same as other accesses.
4	<b>MemoryWriteAndInvalidateCommand: Memory Write and Invalidate Command.</b> Read-write. Reset: 0. 0=Memory Write must be used. 1=Masters may generate the command.
3	<b>SpecialCycle: Special Cycle.</b> Value: 0. 0=Indicates no Special Cycle support.
2	<b>BusMaster: Bus Master.</b> Read-write. Reset: 0. 0=Disable the device from generating PCI accesses. 1=Allow the device to behave as a bus master.
1	<b>MemorySpaceAccesses: Memory Space Accesses.</b> Read-write. Reset: 0. 0=Disable the device response. 1=Allow the device to respond to memory space accesses.
0	<b>IOSpaceAccesses: IO Space Accesses.</b> Read-write. Reset: 0. 0=Disable the device response. 1=Allow the device to respond to IO space accesses.

**D[16,13,12]F2x08 Revision ID / Class Code**

Bits	Description
31:24	<b>BC: Base Class.</b> Value: 0Ch. 0Ch=Identifies the device being a Serial Bus Controller.
23:16	<b>SC: Sub Class.</b> Value: 3h. Sub Class. 03h=Identifies the device being of Universal Serial Bus.
15:8	<b>PI: Programming Interface.</b> Value: 20h. 20h=Identifies the device being an EHCI Host Controller.
7:0	<b>RevisionID: Revision ID.</b> Value: 39h.

**D[16,13,12]F2x0C Miscellaneous**

Bits	Description
31:24	<b>Bist.</b> Value: 0. 00h=BIST is not supported.
23:16	<b>HeaderType: Header Type.</b> Value: 00h. 00h=Indicates EHCI is single function.
15:8	<b>LatencyTimer[7:0]: Latency Timer.</b> Read-only. Reset: 0. This field specifies, in units of PCI bus clocks, the value of the Latency Timer for this PCI bus master.
7:0	<b>CacheLineSize: Cache Line Size.</b> Read-write. Reset: 0. Specifies the system cache line size in units of doublewords .

**D[16,13,12]F2x10 BAR\_EHCI**

Bits	Description
------	-------------

31:8	<b>BA: Base Address[31:8]</b> . Read-write. Reset: 0. Specifies the MMIO Base address of the EHCI USB controller. See 3.25.4.2.2 [EHCI Memory Mapped IO Registers].
7:4	Reserved.
3	<b>PM: Prefetch Memory</b> . Value: 0. 0=Non-prefetchable memory region.
2:1	<b>Type: base address register type</b> . Read-only. Reset: 00b. 00b=32-bit base address register.
0	<b>MemSpace: memory space type</b> . Read-only. Reset: 0. 0=Memory mapped base address.

### D[16,13,12]F2x2C Subsystem ID / Subsystem Vendor ID

Bits	Description
31:16	<b>SubsystemID</b> . Read; Write-once. Reset: 7808h.
15:0	<b>SubsystemVendorID</b> . Read; Write-once. Reset: 1022h.

### D[16,13,12]F2x34 Capability Pointer

Bits	Description
31:8	Reserved.
7:0	<b>CapabilityPointer: Capability Pointer</b> . Read-only. IF (D[16,13,12]F2x50[PMEDisable]==0) THEN Reset: C0h. ELSEIF (D[16,13,12]F2x50[MSIDisable]==0) THEN Reset: D0h. ELSEIF (D[16,13,12]F2x50[FlrEn]==1) THEN Reset: F0h. ELSE Reset: 00h. ENDIF. Specifies the address of the first element of capability link.

### D[16,13,12]F2x3C Interrupt Line

Bits	Description
31:24	<b>MaxLat</b> . Value: 00h.
23:16	<b>MinGnt</b> . Value: 00h.
15:8	<b>InterruptPin: Interrupt Pin</b> . Read-only. Reset: 2h. 02h=INTB#.
7:0	<b>InterruptLine: Interrupt Line</b> . Read-write. Reset: 0. Identifies which input on the interrupt controller the function's PCI interrupt request pin is routed to.

### D[16,13,12]F2x50 EHCI Misc Control

This register is shared among all the EHCI controllers if (D[16,13,12]F0x50[ShareRegisterEnable]==1).

Bits	Description
31	<b>PHYAdvancePowerSavingEnable: PHY Advance Power Saving Enable</b> . Read-write. Reset: 1. 1=Enable the advance PHY power saving feature to save active power from the USB PHY.
30:29	Reserved.
28	<b>DisableAsyncQHCACHEEnhancement: Disable Async QH Cache Enhancement</b> . Read-write. Reset: 0. 1=Disable async QH/QTD cache enhancement.
27	<b>DisablePeriodicListCache: Disable Periodic List Cache</b> . Read-write. Reset: 0. 1=Disable periodic list cache.



26	<b>DisableAsyncDataCache: Disable Async Data Cache.</b> Read-write. Reset: 0. 1=Disable async data cache request.								
25	<b>DisableAsyncQHCACHE: Disable Async QH Cache.</b> Read-write. Reset: 0. 1=Disable async QH/QTD cache.								
24	<b>AsyncQHCACHEThresholdControl: Async QH Cache Threshold Control.</b> Read-write. Reset: 0. For OUT, 0=Cache only if packet size is greater than 128 Bytes. 1=Cache only if packet size is greater than 64 Bytes. For IN, 0=Cache only if expected packet size is greater than 192 Bytes. 1=Cache only if expected packet size is greater than 96 Bytes.								
23	<b>AsyncParkDisable: async-park disable.</b> Read-write. Reset: 0. 1=Disable async-park mode.								
22	Reserved.								
21	<b>EnableIPGapExt: Enable InterPacket Gap Extension.</b> Read-write. Reset: 1. 1=Enable inter-packet gap extension in PIE idle state on starting a new transaction.								
20	Reserved.								
19	<b>QualifyCacheHit: Qualify Cache Hit.</b> Read-write. Reset: 0. BIOS: 1. 1=Qualify cache hit with EHCI_RDWRN command.								
18	Reserved.								
17	<b>AsyncParkCacheControl: Async Park Cache Control.</b> Read-write. Reset: 1. 1=Enable async park cache control.								
16	Reserved.								
15:12	<b>AsyncParkOUTControl: Async Park OUT Control.</b> Read-write. Reset: 1. Async Park Mode Count for OUT Packet. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Standard count as specified by <a href="#">EHCI[3:1]x20[AsyncScheduleParkModeCount]</a>.</td> </tr> <tr> <td>1h</td> <td>8 packets.</td> </tr> <tr> <td>Fh-2h</td> <td>Reserved.</td> </tr> </tbody> </table>	Bits	Definition	0h	Standard count as specified by <a href="#">EHCI[3:1]x20[AsyncScheduleParkModeCount]</a> .	1h	8 packets.	Fh-2h	Reserved.
Bits	Definition								
0h	Standard count as specified by <a href="#">EHCI[3:1]x20[AsyncScheduleParkModeCount]</a> .								
1h	8 packets.								
Fh-2h	Reserved.								
11:8	<b>AsyncParkINControl: Async Park IN Control.</b> Read-write. Reset: 1. Async Park Mode Count for IN Packet.. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Standard count as specified by <a href="#">EHCI[3:1]x20[AsyncScheduleParkModeCount]</a>.</td> </tr> <tr> <td>1h</td> <td>8 packets.</td> </tr> <tr> <td>Fh-2h</td> <td>Reserved.</td> </tr> </tbody> </table>	Bits	Definition	0h	Standard count as specified by <a href="#">EHCI[3:1]x20[AsyncScheduleParkModeCount]</a> .	1h	8 packets.	Fh-2h	Reserved.
Bits	Definition								
0h	Standard count as specified by <a href="#">EHCI[3:1]x20[AsyncScheduleParkModeCount]</a> .								
1h	8 packets.								
Fh-2h	Reserved.								
7	<b>EnablePerPortChangeEventsCapability: Enable Per-Port Change Events Capability.</b> Read-write. Reset: 0. Enable the support of Per-Port Change Event Capability. 1=Set <a href="#">EHCI[3:1]x08[PerPortChangeEventCapability]</a> to 1. 0=Clear <a href="#">EHCI[3:1]x08[PerPortChangeEventCapability]</a> to 0.								
6	<b>MSIDisable: MSI Disable.</b> Read-write. Reset: 0. BIOS: 1. 1=Disable MSI support.								
5	<b>PMEDisable: PME Disable.</b> Read-write. Reset: 0. 1=Disable PME support.								
4	<b>EhciPmeLevelSignalByResumeWakeupEnable: EHCI PME level signal by resume-wakeup enable.</b> Read-write. Reset: 0. 1=PME is a level signal instead of pulse when resume wake-up event is detected.								
3	Reserved.								
2	<b>FlrEn: FLR Enable.</b> Read-write. Reset: 0. 1=Enable FLR support.								
1	<b>DisableSMI: disable SMI.</b> Read-write. Reset: 0. 1=Disable EHCI_SMI sent to USB SMI output (to ACPI). 0=Enable.								

0	<b>D3ColdPMEsupport: D3Cold PME support.</b> Read-write. Reset: 0. BIOS: 1. 1=Enable EHCI host controller PME support at D3Cold. 0=Disable.
---	---------------------------------------------------------------------------------------------------------------------------------------------

**D[16,13,12]F2x54 EHCI Spare 1**

This register is shared among all the EHCI controllers if (D[16,13,12]F0x50[ShareRegisterEnable]==1).

Bits	Description																				
31	Reserved.																				
30:27	Reserved.																				
26	<b>EnMaskIntrOfD0: Enable masking interrupt out of D0 state.</b> Read-write. Reset: 0. 0=Disable masking interrupt for all state, including D3hot. 1=Enable masking interrupt out of D0 state.																				
25:24	Reserved.																				
23	Reserved.																				
22:15	Reserved.																				
14	Reserved																				
13	Reserved.																				
12:11	Reserved.																				
10	<b>ResumePmeD123En: Resume PME in D123 Enable.</b> Read-write. Reset: 0. 1=PME Status will be set only in non-D0 state when resume occurs.																				
9:7	<b>EhciL1EarlyExitTimer: EHCI L1 early exit timer.</b> Read-write. Reset: 5h. BIOS: 100b. Timer to control EHCI L1 exit. When EhciL1EarlyExitEnable is set to 1, this field selects the amount of time before SOF that the EHCI controller will request UMI to exit from L1. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>20 us</td> <td>100b</td> <td>60 us</td> </tr> <tr> <td>001b</td> <td>30 us</td> <td>101b</td> <td>70 us</td> </tr> <tr> <td>010b</td> <td>40 us</td> <td>110b</td> <td>80 us</td> </tr> <tr> <td>011b</td> <td>50 us</td> <td>111b</td> <td>90 us</td> </tr> </tbody> </table>	Bits	Definition	Bits	Definition	000b	20 us	100b	60 us	001b	30 us	101b	70 us	010b	40 us	110b	80 us	011b	50 us	111b	90 us
Bits	Definition	Bits	Definition																		
000b	20 us	100b	60 us																		
001b	30 us	101b	70 us																		
010b	40 us	110b	80 us																		
011b	50 us	111b	90 us																		
6	<b>EhciL1EarlyExitMode: Ehci L1 early exit mode.</b> Read-write. Reset: 0. BIOS: 1. Mode selection bit for EHCI L1 early exit function. 0=Exit signal active in every uFrame if PDC schedule is enabled. 1=Exit signal active in current uFrame when detecting next uFrame has active PDC work.																				
5	<b>EhciL1EarlyExitEnable: EHCI L1 early exit enable.</b> Read-write. Reset: 0. BIOS: 1. 0=Disable. 1=Enable.																				
4	<b>LSConnectionWakeUpEnable: LS connection wake up enable.</b> Read-write. Reset: 1. 1=Enable wake up from S states for Low Speed (LS) connection. 0=Disable.																				
3	<b>EhciEmptyListMode: EHCI empty list mode.</b> Read-write. Reset: 1. 1=Enable empty list mode. 0=Disable.																				
2	<b>EhciBlockCStateEnable: EHCI block-C-state enable.</b> Read-write. Reset: 0. 1=Enable EHCI sending block-C-state signal to ACPI.																				
1	Reserved.																				
0	<b>EhciDelayBifL1Enable: EHCI delay BIF L1 enable.</b> Read-write. Reset: 0. 1=Enable EHCI delaying BIF on entering L1 when there are USB transactions on the controller and when USB delaying BIF L1 is enabled in A-link bridge.																				

**D[16,13,12]F2x60 Serial Bus Release Number / FLADJ**

Bits	Description
31:14	Reserved.
13:8	<b>FLADJ: Frame Length Timing Value.</b> Read-write. Reset: 20h. Each decimal value change to this register corresponds to 16 high-speed bit times. The SOF cycle time (number of SOF counter clock periods to generate a SOF micro-frame length)=59488 + 16*FLAdj.
7:0	<b>SBRN: Serial Bus Release Number.</b> Value: 20h.

**D[16,13,12]F2xA0 USB Legacy Support Extended Capability**

Bits	Description
31:25	Reserved.
24	<b>HcOsOwnedSemaphore.</b> Read-write. Reset: 0. System software sets this bit to request ownership of the EHCI controller. Ownership is obtained when this bit reads as 1 and the HcBiosOwnedSemaphore reads as 0.
23:17	Reserved.
16	<b>HcBiosOwnedSemaphore.</b> Read-write. Reset: 0. The BIOS sets this bit to establish ownership of the EHCI controller. System BIOS will set this bit to a 0 in response to a request for ownership of the EHCI controller by system software.
15:8	<b>NextEhciExtendedCapabilityPointer.</b> Read-only. Reset: 00h. 00h=The end of the extended capability list.
7:0	<b>CapabilityID.</b> Read-only. Reset: 01h. 01h=Legacy Support.

**D[16,13,12]F2xA4 USB Legacy Support Control / Status**

Bits	Description
31	<b>SmiOnBar.</b> Read; Set-by-hardware; write-1-to-clear. Reset: 0. 1=The Base Address Register (BAR) is written.
30	<b>SmiOnPciCommand.</b> Read; Set-by-hardware; write-1-to-clear. Reset: 0. 1=The PCI Command Register is written.
29	<b>SmiOnOsOwnershipChange.</b> Read; set-by-hardware; write-1-to-clear. Reset: 0. 1=D[16,13,12]F2xA0[HcOsOwnedSemaphore] transitions from 1 to 0 or 0 to 1.
28:22	Reserved.
21	<b>SmiOnAsyncAdvance.</b> Read-only. Reset: 0. This is the shadow bit of EHCI[3:1]x24[InterruptOnAsyncAdvance]. Writing 1 to EHCI[3:1]x24[InterruptOnAsyncAdvance] clears this bit to 0.
20	<b>SmiOnHostSystemError.</b> Read-only. Reset: 0. This is the shadow bit of EHCI[3:1]x24[HostSystemError]. Writing 1 to EHCI[3:1]x24[HostSystemError] clears this bit to 0.
19	<b>SmiOnFrameListRollover.</b> Read-only. Reset: 0. This is the shadow bit of EHCI[3:1]x24[FrameListRollover]. Writing 1 to EHCI[3:1]x24[FrameListRollover] clears this bit to 0.
18	<b>SmiOnPortChangeDetect.</b> Read-only. Reset: 0. This is the shadow bit of EHCI[3:1]x24[PortChangeDetect]. Writing 1 to EHCI[3:1]x24[PortChangeDetect] clears this bit to 0.

17	<b>SmiOnUsbError.</b> Read-only. Reset: 0. This is the shadow bit of <b>EHCI[3:1]x24[UsbErrInt]</b> . Writing 1 to <b>EHCI[3:1]x24[UsbErrInt]</b> clears this bit to 0.
16	<b>SmiOnUsbComplete.</b> Read-only. Reset: 0. This is the shadow bit of <b>EHCI[3:1]x24[UsbInt]</b> . Writing 1 to <b>EHCI[3:1]x24[UsbInt]</b> clears this bit to 0.
15	<b>SmiOnBarEnable.</b> Read-write. Reset: 0. 1=The host controller will issue an SMI if <b>SmiOnBAR==1</b> .
14	<b>SmiOnPciCommandEnable.</b> Read-write. Reset: 0. 1=The host controller will issue an SMI if <b>SmiOnPciCommand==1</b> .
13	<b>SmiOnOsOwnershipEnable.</b> Read-write. Reset: 0. 1=The host controller will issue an SMI if <b>SmiOnOsOwnershipChange==1</b> .
12:6	Reserved.
5	<b>SmiOnAsyncAdvanceEnable.</b> Read-write. Reset: 0. 1=The host controller will issue an SMI if <b>SmiOnAsyncAdvance==1</b> .
4	<b>SmiOnHostSystemErrorEnable.</b> Read-write. Reset: 0. 1=The host controller will issue an SMI if <b>SmiOnHostSystemError==1</b> .
3	<b>SmiOnFrameListRolloverEnable.</b> Read-write. Reset: 0. 1=The host controller will issue an SMI if <b>SmiOnFrameListRollover==1</b> .
2	<b>SmiOnPortChangeEnable.</b> Read-write. Reset: 0. 1=The host controller will issue an SMI if <b>SmiOnPortChangeDetect==1</b> .
1	<b>SmiOnUsbErrorEnable.</b> Read-write. Reset: 0. 1=The host controller will issue an SMI if <b>SmiOnUsbError==1</b> .
0	<b>UsbSmiEnable.</b> Read-write. Reset: 0. 1=The host controller will issue an SMI if <b>SmiOnUsbComplete==1</b> .

**D[16,13,12]F2xC0 PME Capability**

Bits	Description												
31:27	<b>PmeSupport[4:0].</b> Read-only. Reset: 0Fh. Indicates the power states in which the function may assert PME#. For each bit, 0=the function is not capable of asserting the PME# signal while in that power state. <table border="0" style="margin-left: 20px;"> <tr> <td style="text-align: right;"><u>Bit</u></td> <td style="text-align: left;"><u>Definition</u></td> </tr> <tr> <td style="text-align: right;">[0]</td> <td>1=PME# can be asserted from D0</td> </tr> <tr> <td style="text-align: right;">[1]</td> <td>1=PME# can be asserted from D1</td> </tr> <tr> <td style="text-align: right;">[2]</td> <td>1=PME# can be asserted from D2</td> </tr> <tr> <td style="text-align: right;">[3]</td> <td>1=PME# can be asserted from D3hot</td> </tr> <tr> <td style="text-align: right;">[4]</td> <td>1=PME# can be asserted from D3cold</td> </tr> </table>	<u>Bit</u>	<u>Definition</u>	[0]	1=PME# can be asserted from D0	[1]	1=PME# can be asserted from D1	[2]	1=PME# can be asserted from D2	[3]	1=PME# can be asserted from D3hot	[4]	1=PME# can be asserted from D3cold
<u>Bit</u>	<u>Definition</u>												
[0]	1=PME# can be asserted from D0												
[1]	1=PME# can be asserted from D1												
[2]	1=PME# can be asserted from D2												
[3]	1=PME# can be asserted from D3hot												
[4]	1=PME# can be asserted from D3cold												
26	<b>D2Support.</b> Read-only. Reset: 1. 1=This function supports the D2 Power Management State.												
25	<b>D1Support.</b> Read-only. Reset: 1. 1=This function supports the D1 Power Management State.												
24:22	<b>AuxCurrent.</b> Value: 0. Reports the 3.3V auxiliary current requirements for the PCI function. 0=Self-powered.												
21	<b>DSI: device specific initialization.</b> Value: 0. Indicates whether special initialization of this function is required before the generic class device driver is able to use it.												
20	Reserved.												
19	<b>PmeClock: PME clock.</b> Value: 0. 0=Indicates that no PCI clock is required for the function to generate PME#.												

18:16	<b>Version.</b> Read-only. Reset: 2h. 2h=Indicates that this function complies with Revision 1.1 of the PCI Power Management Interface Specification.
15:8	<b>NextItemPointer: next item pointer.</b> Read-only. Reset: D0h. Specifies the address of the next capability structure.
7:0	<b>CapId: capability ID.</b> Value: 01h. Identifies the linked list items as being the PCI Power Management registers.

**D[16,13,12]F2xC4 PME Control / Status**

Bits	Description												
31:16	Reserved.												
15	<b>PmeStatus.</b> Read; set-by-hardware; write-1-to-clear. Cold reset: 0. 1=PME# signal is asserted, independent of the state of the PmeEn bit.												
14:9	Reserved.												
8	<b>PmeEn.</b> Read-write. Cold reset: 0. 1=Enables the function to assert PME#. 0=PME# assertion is disabled.												
7:2	Reserved.												
1:0	<b>PowerState.</b> Read-write. Reset: 0. This 2-bit field is used both to determine the current power state of a function and to set the function into a new power state. If software attempts to write an unsupported, optional state to this field, the write operation must be completed normally on the bus; however, the data is discarded and no state change occurs. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Bits</th> <th>Definition</th> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>D0</td> <td>10b</td> <td>D2</td> </tr> <tr> <td>01b</td> <td>D1</td> <td>11b</td> <td>D3hot</td> </tr> </tbody> </table>	Bits	Definition	Bits	Definition	00b	D0	10b	D2	01b	D1	11b	D3hot
Bits	Definition	Bits	Definition										
00b	D0	10b	D2										
01b	D1	11b	D3hot										

**D[16,13,12]F2xD0 MSI Control**

Bits	Description
31:24	Reserved.
23	<b>C64: 64-bit Address Capable.</b> Read-only. IF (EHCI[3:1]x08[AddressingCapability64Bit]==1) THEN Reset: 1. ELSE Reset: 0. ENDIF. 1=Indicates that the EHCI is capable of generating a 64-bit message address. 0=Indicates the EHCI is not capable of generating a 64-bit address.
22:20	<b>MSIControl: MSI Control.</b> Read-write. Reset: 0.
19:17	Reserved.
16	<b>MSIE: MSI Enable.</b> Read-write. Reset: 0. 1=MSI is enabled and the traditional pins are not used to generate interrupts. 0=MSI operation is disabled and the traditional interrupt pins are used.
15:8	<b>NextItemPointer: Next Item Pointer.</b> Read-only. Reset: E4h. Pointer to next capability structure.
7:0	<b>CapabilityID.</b> Read-only. Reset: 5h. Indicates this is USB MSI capability ID.

**D[16,13,12]F2xD4 MSI Address**

Bits	Description

31:2	<b>MsiAddress: MSI Address.</b> Read-write. Reset: 0. Specifies system specified message address [31:2].
1:0	Reserved.

### **D[16,13,12]F2xD8 MSI Upper Address**

Bits	Description
31:0	<b>MsiUpperAddress: MSI Upper Address.</b> Read-write. Reset: 0. IF (D[16,13,12]F2xD0[C64]==1) THEN This field specifies the upper 32 bit of system specific message address [63:32]. ELSE This field specifies the MSI data. ENDIF.

### **D[16,13,12]F2xDC MSI Data**

Bits	Description
31:0	<b>MsiData: MSI Data.</b> Read-write. Reset: 0. IF (D[16,13,12]F2xD0[C64]==1) THEN This field specifies the MSI data. ELSE This field is reserved. ENDIF.

### **D[16,13,12]F2xF0 Function Level Reset Capability**

IF (D[16,13,12]F2x50[FlrEn]==0) THEN

Bits	Description
31:0	Reserved.

ELSE

Bits	Description
31:26	Reserved.
25	<b>FunctionLevelResetCapability: Function Level Reset Capability.</b> Value: 1. 1=Function level reset support.
24	<b>TransactionPendingCapability.</b> Value: 1. 1=Transaction pending feature is supported.
23:16	<b>Length.</b> Read-only. Reset: 6h. Advanced Feature (AF) structure length (byte).
15:8	<b>NextItemPointer: Next Item Pointer.</b> Value: 0. Pointer to next capability structure. 0=This is the final item on the list.
7:0	<b>CapId.</b> Read-only. Reset: 13h. 13h=Identifies that the function is Advanced Feature (AF) capable.

ENDIF.

### **D[16,13,12]F2xF4 Function Level Reset Control**

IF (D[16,13,12]F2x50[FlrEn]==0) THEN

Bits	Description
31:0	Reserved.

ELSE

Bits	Description
31:9	Reserved.
8	<b>TransactionPending.</b> Read-only; updated-by-hardware. Reset: 0. 1=Indicates that the Function has issued one or more non-posted transactions which have not been completed, including non-posted transactions that a target has terminated with Retry. 0=Indicates that all non-posted transactions have been completed.
7:1	Reserved.
0	<b>InitiateFLR: Initiate FLR.</b> RAZ; write-1-only. Reset: 0. 1=Initiates Function Level Reset (FLR).

ENDIF.

### 3.25.4.2.2 EHCI Memory Mapped IO Registers

The EHCI MMIO base address is specified by  $D[16,13,12]F2x10[BA]$ . For details of ECHI USB controller implementation requirements and behavior consult the *Universal Serial Bus Specification 2.0* and *Enhanced Host Controller Interface Specification for Universal Serial Bus*.

#### **EHCI[3:1]x00 Capability Length**

Bits	Description
7:0	<b>CapLength.</b> Value: 20h. This register is used as an offset to add to the register base to find the beginning of the Operational Register Space starting at $EHCI[3:1]x20$ .

#### **EHCI[3:1]x02 HC Interface Version**

Bits	Description
15:0	<b>HciVersion.</b> Value: 110h. Specifies a BCD encoding of the version number of interface to which this host controller interface conforms.

#### **EHCI[3:1]x04 HC Structural Parameters**

**Table 234: Reset mapping for  $EHCI[3:1]x04$**

Register	Reset
$EHCI1x04$	0010_1404h
$EHCI2x04$	0010_1404h
$EHCI3x04$	0010_1202h

Bits	Description
31:24	Reserved.
23:20	<b>DebugPortNumber.</b> Read-only. Reset: 1. This register identifies which of the host controller ports is the debug port. The value is the port number (one-based) of the debug port. A non-zero value in this field indicates the presence of a debug port. The value in this register must not be greater than $EHCI[3:1]x04[NPorts]$ .
19:17	Reserved.

16	<b>PortIndicators.</b> Read-only. Indicates whether the ports support port indicator control. 1=The port status and control registers include a read/writable field for controlling the state of the port indicator.						
15:12	<b>N_CC: number of companion controllers.</b> Read-only. Indicates the number of companion controllers associated with this USB 2.0 host controller. A value larger than 0 in this field indicates there are companion USB 1.1 host controller(s). Port-ownership handoffs are supported. High-speed, Full-speed and Low-speed devices are supported on the host controller root ports.						
11:8	<b>N_PCC: NumberOfPortsPerCompanionController.</b> Read-only. Indicates the number of ports supported per companion host controller. It is used to indicate the port routing configuration to system software.						
7	<b>PortRoutingRules.</b> Read-only. Specifies how all ports are mapped to companion controllers. 0=The first N_PCC ports are routed to the lowest numbered function companion host controller, the next N_PCC port are routed to the next lowest function companion controller, and so on.						
6:5	Reserved.						
4	<b>PPC: PortPowerControl.</b> Read-only. Specifies whether the host controller implementation includes port power control. 0=The port does not have port power switches. The value of this field affects the functionality of <a href="#">EHCI[3:1]x[70,6C,68,64][PortPower]</a> .						
3:0	<b>NPorts.</b> Read-only. Specifies the number of physical downstream ports implemented on this host controller. The value of this field determines how many port registers are addressable in the Operational Register Space. <table border="0"> <tr> <td><u>Bits</u></td> <td><u>Definition</u></td> </tr> <tr> <td>0h</td> <td>undefined</td> </tr> <tr> <td>Fh-1h</td> <td>&lt;NPorts&gt; Ports</td> </tr> </table>	<u>Bits</u>	<u>Definition</u>	0h	undefined	Fh-1h	<NPorts> Ports
<u>Bits</u>	<u>Definition</u>						
0h	undefined						
Fh-1h	<NPorts> Ports						

### **EHCI[3:1]x08 HC Capability Parameters**

Bits	Description
31:20	Reserved.
19	<b>Frame32PeriodicListCapability: 32-Frame Periodic List Capability.</b> Read-only. Reset: 0. 0=Software must treat a Frame List Size value of 11b as reserved.
18	<b>PerPortChangeEventCapability: Per-Port Change Event Capability.</b> Value: <a href="#">D[16,13,12]F2x50[EnablePerPortChangeEventsCapability]</a> . 1=Host controller supports per-port change events and the associated fields <a href="#">EHCI[3:1]x20[PerPortChangeEventsEnable]</a> , <a href="#">EHCI[3:1]x24[PortNChangeDetect]</a> and <a href="#">EHCI[3:1]x28[PortChangeIntEn]</a> . 0=Per-port change events are not supported; software should treat those fields as reserved.
17	<b>LinkPowerManagementCapability: Link Power Management Capability.</b> Read-only. Reset: 0. 0=Link Power Management L1 State is not supported.
16	<b>HardwarePrefetchCapability: Hardware Prefetch Capability.</b> Read-only. Reset: 0. 0=Hardware prefetch capability is not supported.
15:8	<b>EhciExtendedCapabilitiesPointer.</b> Read-only. Reset: A0h. Specifies the offset in PCI configuration space of the first EHCI extended capability. See <a href="#">D[16,13,12]F2xA0</a> .
7	<b>IsochronousSchedulingThreshold3.</b> Read-only. Reset: 0. The value of this field determines the function of <a href="#">IsochronousSchedulingThreshold[2:0]</a> .



6:4	<b>IsynchronousSchedulingThreshold[2:0]</b> . Read-only. Reset: 111b. This field indicates, relative to the current position of the executing host controller, where software can reliably update the isochronous schedule. IF (IsynchronousSchedulingThreshold3==0) THEN Field value indicates the number of micro-frames of isochronous data structures (one or more) a host controller can hold before flushing the state. ELSE Host software assumes the host controller may cache an isochronous data structure for an entire frame. ENDIF.
3	Reserved.
2	<b>AsyncScheduleParkCapability</b> . Read-only. Reset: 1. 1=The host controller supports the park feature for high-speed queue heads in the Asynchronous Schedule.
1	<b>ProgrammableFrameListFlag</b> . Read-only. Reset: 1. 0= <b>EHCI[3:1]x20[FrameListSize]</b> is read-only; System software must use a frame list length of 1024 elements with this host controller. 1= <b>EHCI[3:1]x20[FrameListSize]</b> is read-writeable; System software can specify and use a smaller frame list via <b>EHCI[3:1]x20[FrameListSize]</b> .
0	<b>AddressingCapability64Bit</b> . Read-only. Reset: 0. Specifies the addressing range capability of this implementation. 0=Data structures using 32-bit address memory pointers. 1=Data structures using 64-bit address memory pointers.

**EHCI[3:1]x20 USB Command**

Bits	Description																				
31:28	Reserved.																				
27:24	<b>HostInitiatedResumeDuration: host-initiated resume duration</b> . Read-only. Reset: 0. This field is used by system software to specify the minimum amount of time the host controller will drive the K-state during a host-initiated resume from a LPM state (e.g. L1), and is conveyed to each LPM-enabled device (via the HIRD bits within an LPM Token’s bmAttributes field) upon entry into a low-power state.  <table border="0"> <tr> <td><u>Bits</u></td> <td><u>Definition</u></td> </tr> <tr> <td>Fh-0h</td> <td>(&lt;HostInitiatedResumeDuration&gt; *75 us) + 50 us</td> </tr> </table>	<u>Bits</u>	<u>Definition</u>	Fh-0h	(<HostInitiatedResumeDuration> *75 us) + 50 us																
<u>Bits</u>	<u>Definition</u>																				
Fh-0h	(<HostInitiatedResumeDuration> *75 us) + 50 us																				
23:16	<b>InterruptThresholdControl: Interrupt Threshold Control</b> . Read-write. Reset: 8h. This field is used by system software to select the maximum rate at which the host controller will issue interrupts. The only valid values are defined below. Any other value in this register yields undefined results.  <table border="0"> <tr> <td><u>Bits</u></td> <td><u>Definition</u></td> <td><u>Bits</u></td> <td><u>Definition</u></td> </tr> <tr> <td>00h</td> <td>Reserved</td> <td>08h</td> <td>8 micro-frames (default, 1 ms)</td> </tr> <tr> <td>01h</td> <td>1 micro-frame</td> <td>10h</td> <td>16 micro-frames (2 ms)</td> </tr> <tr> <td>02h</td> <td>2 micro-frames</td> <td>20h</td> <td>32 micro-frames (4 ms)</td> </tr> <tr> <td>04h</td> <td>4 micro-frames</td> <td>40h</td> <td>64 micro-frames (8 ms)</td> </tr> </table> Software modifications to this bit while ( <b>EHCI[3:1]x24[HCHalted]</b> ==0) results in undefined behavior.	<u>Bits</u>	<u>Definition</u>	<u>Bits</u>	<u>Definition</u>	00h	Reserved	08h	8 micro-frames (default, 1 ms)	01h	1 micro-frame	10h	16 micro-frames (2 ms)	02h	2 micro-frames	20h	32 micro-frames (4 ms)	04h	4 micro-frames	40h	64 micro-frames (8 ms)
<u>Bits</u>	<u>Definition</u>	<u>Bits</u>	<u>Definition</u>																		
00h	Reserved	08h	8 micro-frames (default, 1 ms)																		
01h	1 micro-frame	10h	16 micro-frames (2 ms)																		
02h	2 micro-frames	20h	32 micro-frames (4 ms)																		
04h	4 micro-frames	40h	64 micro-frames (8 ms)																		
15	<b>PerPortChangeEventEnable: Per-Port Change Events Enable</b> . IF ( <b>EHCI[3:1]x08[PerPortChangeEventCapability]</b> ==1) THEN Read-write. ELSE Read-only. ENDIF. Reset: 0. 1=Enable the per-port change event capability as defined by <b>EHCI[3:1]x24[PortNChangeDetect]</b> and <b>EHCI[3:1]x28[PortNChangeEventEnable]</b> .																				
14	<b>FullySynchronizedPrefetch: Fully Synchronized Prefetch</b> . Read-only. Reset: 0.																				
13	<b>AsynchronousSchedulePrefetchEnable: Asynchronous Schedule Prefetch Enable</b> . Read-only. Reset: 0.																				
12	<b>PeriodicSchedulePrefetchEnable: Periodic Schedule Prefetch Enable</b> . Read-only. Reset: 0.																				

11	<b>AsyncScheduleParkModeEnable: asynchronous schedule park mode enable.</b> Read-write. Reset: 1. 1=Park mode is enabled. 0=Park mode is disabled.												
10	Reserved.												
9:8	<b>AsyncScheduleParkModeCount: Asynchronous Schedule Park Mode Count.</b> Read-write. Reset: 11b. This field contains a count of the number of successive transactions the host controller is allowed to execute from a high-speed queue head on the Asynchronous schedule before continuing traversal of the Asynchronous schedule. Valid values are 1h to 3h. Software must not write a 0 to this field when AsyncScheduleParkModeEnable==1, as this will result in undefined behavior.												
7	<b>LightHostControllerReset: Light Host Controller Reset.</b> Read-only. Reset: 0. Not implemented.												
6	<b>InterruptOnAsyncAdvanceDoorbell: Interrupt on Async Advance Doorbell.</b> Read; cleared-by-hardware; write-1-only. Reset: 0. This bit is used as a doorbell by software to tell the host controller to issue an interrupt the next time it advances asynchronous schedule. Software must write a 1 to this bit to ring the doorbell. When the host controller has evicted all appropriate cached schedule state, it sets <a href="#">EHCI[3:1]x24[InterruptOnAsyncAdvance]</a> to 1 and clears this bit to 0. Software should not write a 1 to this bit when the asynchronous schedule is disabled. Doing so will yield undefined results.												
5	<b>AsynchronousScheduleEnable: Asynchronous Schedule Enable.</b> Read-write. Reset: 0. This bit controls whether the host controller skips processing the Asynchronous Schedule. 0=Do not process the Asynchronous Schedule. 1=Use <a href="#">EHCI[3:1]x38</a> to access Asynchronous Schedule.												
4	<b>PeriodicScheduleEnable: Periodic Schedule Enable.</b> Read-write. Reset: 0. This bit controls whether the host controller skips processing the Periodic Schedule. 0=Do not process the Periodic Schedule. 1=Use <a href="#">EHCI[3:1]x34</a> to access the Periodic Schedule.												
3:2	<b>FrameListSize: frame list size.</b> IF (( <a href="#">EHCI[3:1]x08[ProgrammableFrameListFlag]</a> ==1)    ( <a href="#">EHCI[3:1]x08[Frame32PeriodicListCapability]</a> ==1)) THEN Read-write. ELSE RAZ. ENDIF. Reset: 0. This field specifies the size of the frame list. The size the frame list controls which bits in the Frame Index Register should be used for the Frame List Current index. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Bits</th> <th>Definition</th> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>1024 frames (4096 bytes)</td> <td>01b</td> <td>512 frames (2098 bytes)</td> </tr> <tr> <td>10b</td> <td>256 frames (1024 bytes)</td> <td>11b</td> <td>Reserved or 32 frames (see below)</td> </tr> </tbody> </table> Software may only program a 32 frame list when <a href="#">EHCI[3:1]x08[Frame32PeriodicListCapability]</a> =1, and a 512 or 256 frame list when <a href="#">EHCI[3:1]x08[ProgrammableFrameListFlag]</a> =1.	Bits	Definition	Bits	Definition	00b	1024 frames (4096 bytes)	01b	512 frames (2098 bytes)	10b	256 frames (1024 bytes)	11b	Reserved or 32 frames (see below)
Bits	Definition	Bits	Definition										
00b	1024 frames (4096 bytes)	01b	512 frames (2098 bytes)										
10b	256 frames (1024 bytes)	11b	Reserved or 32 frames (see below)										
1	<b>HCRESET: Host Controller Reset.</b> Read; cleared-by-hardware; write-1-only. Reset: 0. This control bit is used by software to reset the host controller. The effects of this on Root Hub registers are similar to a Chip Hardware Reset. Software must reinitialize the host controller in order to return the host controller to an operational state. 1=The Host Controller resets its internal pipelines, timers, counters, state machines, etc. to their initial value; Any transaction currently in progress on USB is immediately terminated; A USB reset is not driven on downstream ports; PCI Configuration registers are not affected by this reset; All operational registers, including port registers and port state machines are set to their initial values; Port ownership reverts to the companion host controller(s). Software should not set this bit to a 1 when <a href="#">EHCI[3:1]x24[HCHalted]</a> ==0. Attempting to reset an actively running host controller will result in undefined behavior.												

0	<p><b>RunStop: Run/Stop.</b> Read-write. Reset: 0. 1=Run; the Host Controller proceeds with execution of the schedule as long as this bit is set to 1. 0=Stop; the Host Controller completes the current and any actively pipelined transactions on the USB and then halts within 16 micro-frames after software clears this bit.</p> <p>Software must not write a 1 to this field unless <code>EHCI[3:1]x24[HCHalted]==1</code>. Doing so will yield undefined results.</p>
---	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**EHCI[3:1]x24 USB Status**

Bits	Description										
31:16	<p><b>PortNChangeDetect: Port-n Change Detect.</b> IF (<code>EHCI[3:1]x08[PerPortChangeEventCapability]==1</code>) THEN Read; set-by-hardware; write-1-to-clear. ELSE Read-only. ENDIF. Reset: 0. This field should only be used by software when (<code>EHCI[3:1]x20[PerPortChangeEventsEnable]==1</code>). Each bit in this field correspond to a port. For each bit, 1=A port change event was detected for that port; see <code>EHCI[3:1]x24[PortChangeDetect]</code> for port change event details.</p> <p><code>EHCI[3:1]x04[NPorts]</code> specifies how many ports are exposed by the host controller and thus how many bits in this field are valid.</p> <table border="0" style="margin-left: 20px;"> <tr> <td style="text-align: center;"><u>Bit</u></td> <td style="text-align: center;"><u>Definition</u></td> </tr> <tr> <td style="text-align: center;">[0]</td> <td style="text-align: center;">Port 1</td> </tr> <tr> <td style="text-align: center;">[1]</td> <td style="text-align: center;">Port 2</td> </tr> <tr> <td style="text-align: center;">...</td> <td style="text-align: center;">...</td> </tr> <tr> <td style="text-align: center;">[15]</td> <td style="text-align: center;">Port 16</td> </tr> </table>	<u>Bit</u>	<u>Definition</u>	[0]	Port 1	[1]	Port 2	...	...	[15]	Port 16
<u>Bit</u>	<u>Definition</u>										
[0]	Port 1										
[1]	Port 2										
...	...										
[15]	Port 16										
15	<p><b>AsynchronousScheduleStatus: Asynchronous Schedule Status.</b> Read-only. Reset: 0. The bit reports the current real status of the Asynchronous Schedule. 0=Status of the Asynchronous Schedule is disabled. 1=Status of the Asynchronous Schedule is enabled.</p> <p>The host controller is not required to immediately disable or enable the Asynchronous Schedule when software transitions <code>EHCI[3:1]x20[AsynchronousScheduleEnable]</code>. When this bit and the <code>EHCI[3:1]x20[AsynchronousScheduleEnable]</code> are the same value, the Asynchronous Schedule is either enabled (1) or disabled (0).</p>										
14	<p><b>PeriodicScheduleStatus: Periodic Schedule Status.</b> Read-only. Reset: 0. The bit reports the current real status of the Periodic Schedule. 0=Periodic Schedule is disabled. 1=Periodic Schedule is enabled.</p> <p>The host controller is not required to immediately disable or enable the Periodic Schedule when software transitions <code>EHCI[3:1]x20[PeriodicScheduleEnable]</code>. When this bit and the Periodic Schedule Enable bit are the same value, Periodic Schedule is either enabled (1) or disabled (0).</p>										
13	<p><b>Reclamation.</b> Read-only. Reset: 0. Indicates an empty asynchronous schedule.</p>										
12	<p><b>HCHalted.</b> Read-only. Reset: 1. 0=<code>EHCI[3:1]x20[RunStop]</code> is 1 or the host controller has not yet stopped. 1=The host controller has stopped executing as a result of <code>EHCI[3:1]x20[RunStop]</code> being set to 0, either by software or by the host controller hardware due to an internal error.</p>										
11:6	Reserved.										
5	<p><b>InterruptOnAsyncAdvance: Interrupt on Async Advance.</b> Read-write. Reset: 0. 1=Indicate the assertion of interrupt source caused by <code>EHCI[3:1]x20[InterruptOnAsyncAdvanceDoorbell]</code>.</p>										
4	<p><b>HostSystemError: Host System Error.</b> Read; set-by-hardware; write-1-to-clear. Reset: 0. 1=A serious error occurs during a host system access involving the host controller module. In a PCI system, conditions that set this bit to 1 include PCI Parity error, PCI Master Abort, and PCI Target Abort.</p> <p>When this error occurs, the host controller clears <code>EHCI[3:1]x20[RunStop]</code> to prevent further execution of the scheduled TDs.</p>										

3	<b>FrameListRollover: Frame List Rollover.</b> Read; set-by-hardware; write-1-to-clear. Reset: 0. 1= $\text{EHCI}[3:1]\text{x}2\text{C}[\text{FrameIndex}]$ rolls over from its maximum value to zero. The exact Frame List Index value at which the rollover occurs depends on the frame list size.
2	<b>PortChangeDetect: Port Change Detect.</b> Read; set-by-hardware; write-1-to-clear. Reset: 0. 1=One of the following port change event has occurred: <ul style="list-style-type: none"> <li>• <math>\text{EHCI}[3:1]\text{x}[70,6\text{C},68,64][\text{PortOwner}]</math> changed from 0 to 1;</li> <li>• <math>\text{EHCI}[3:1]\text{x}[70,6\text{C},68,64][\text{ForcePortResume}]</math> changed from 0 to 1;</li> <li>• <math>\text{EHCI}[3:1]\text{x}[70,6\text{C},68,64][\text{ConnectStatusChange}]</math> is set to 1.</li> </ul>
1	<b>UsbErrInt: USB Error Interrupt.</b> Read; set-by-hardware; write-1-to-clear. Reset: 0. 1=The completion of a USB transaction results in an error condition (e.g., error counter underflow). If the TD on which the error interrupt occurred also had its IOC bit set, both this bit and the UsbInt bit are set.
0	<b>UsbInt: USB Interrupt.</b> Read; set-by-hardware; write-1-to-clear. Reset: 0. 1=A USB transaction has completed, which results in the retirement of a Transfer Descriptor that had its IOC bit set; A short packet is detected (actual number of bytes received was less than the expected number of bytes).

**EHCI[3:1]x28 USB Interrupt Enable**

Bits	Description										
31:16	<b>PortNChangeEventEnable: Port-n Change Event Enable.</b> Read-write. Reset: 0. This field is bit significant. Each bit corresponds to a port. For each bit, 1=The host controller will issue an interrupt if the corresponding bit in $\text{EHCI}[3:1]\text{x}24[\text{PortNChangeEvent}]$ is set to 1. The interrupt is acknowledged by software clearing $\text{EHCI}[3:1]\text{x}24[\text{PortNChangeEvent}]$ . $\text{EHCI}[3:1]\text{x}04[\text{NPorts}]$ specifies how many ports are exposed by the host controller and thus how many bits in this field are valid. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Bit</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>[0]</td> <td>Port 1</td> </tr> <tr> <td>[1]</td> <td>Port 2</td> </tr> <tr> <td>...</td> <td>...</td> </tr> <tr> <td>[15]</td> <td>Port 16</td> </tr> </tbody> </table>	Bit	Definition	[0]	Port 1	[1]	Port 2	...	...	[15]	Port 16
Bit	Definition										
[0]	Port 1										
[1]	Port 2										
...	...										
[15]	Port 16										
15:6	Reserved.										
5	<b>InterruptOnAsyncAdvanceEn: Interrupt on Async Advance Enable.</b> Read-write. Reset: 0. 1=The host controller will issue an interrupt at the next interrupt threshold if $\text{EHCI}[3:1]\text{x}24[\text{InterruptOnAsyncAdvance}]=1$ . The interrupt is acknowledged by software clearing $\text{EHCI}[3:1]\text{x}24[\text{InterruptOnAsyncAdvance}]$ .										
4	<b>HostSystemErrorEn: Host System Error Enable.</b> Read-write. Reset: 0. 1=The host controller will issue an interrupt if $\text{EHCI}[3:1]\text{x}24[\text{HostSystemError}]=1$ . The interrupt is acknowledged by software clearing $\text{EHCI}[3:1]\text{x}24[\text{HostSystemError}]$ .										
3	<b>FrameListRolloverEn: Frame List Rollover Enable.</b> Read-write. Reset: 0. 1=The host controller will issue an interrupt if $\text{EHCI}[3:1]\text{x}24[\text{FrameListRollover}]=1$ . The interrupt is acknowledged by software clearing $\text{EHCI}[3:1]\text{x}24[\text{FrameListRollover}]$ .										
2	<b>PortChangeIntEn: Port Change Interrupt Enable.</b> Read-write. Reset: 0. 1=The host controller will issue an interrupt if $\text{EHCI}[3:1]\text{x}24[\text{PortChangeDetect}]=1$ . The interrupt is acknowledged by software clearing $\text{EHCI}[3:1]\text{x}24[\text{PortChangeDetect}]$ .										
1	<b>UsbErrIntEn: USB Error Interrupt Enable.</b> Read-write. Reset: 0. 1=The host controller will issue an interrupt at the next interrupt threshold if $\text{EHCI}[3:1]\text{x}24[\text{UsbErrInt}]=1$ . The interrupt is acknowledged by software clearing $\text{EHCI}[3:1]\text{x}24[\text{UsbErrInt}]$ .										

0	<b>UsbIntEn: USB Interrupt Enable.</b> Read-write. Reset: 0. 1=The host controller will issue an interrupt at the next interrupt threshold if <code>EHCI[3:1]x24[UsbInt]==1</code> . The interrupt is acknowledged by software clearing <code>EHCI[3:1]x24[UsbInt]</code> .
---	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### EHCI[3:1]x2C Frame Index

Bits	Description
31:14	Reserved.
13:0	<b>FrameIndex: Frame Index.</b> Read-write; updated-by-hardware. Reset: 0. This register is used by the host controller to index into the periodic frame list. The register updates every 125 microseconds (once each micro-frame). Bits [N:3] are used to select a particular entry in the Periodic Frame List during periodic schedule execution. The number of bits used for the index depends on <code>EHCI[3:1]x20[FrameListSize]</code> . This register must be written as a DWord. Byte writes produce undefined results. This register cannot be written unless <code>EHCI[3:1]x24[HCHalted]==1</code> . Writes to this register also affect the SOF value.

### EHCI[3:1]x30 Control Data Structure Segment

Bits	Description
31:0	<b>CtrlDSSegment.</b> IF ( <code>EHCI[3:1]x08[AddressingCapability64Bit]==0</code> ) THEN RAZ. ELSE Read-write. ENDIF. Reset: 0. Specifies the most significant address bits [63:32] for all EHCI data structures if <code>EHCI[3:1]x08[AddressingCapability64Bit]==1</code> . This register is concatenated with the link pointer from either <code>EHCI[3:1]x34</code> , or <code>EHCI[3:1]x38</code> , or any control data structure link field to construct a 64-bit address.

### EHCI[3:1]x34 Periodic Frame List Base Address

Bits	Description
31:12	<b>BaseAddress: Base Address.</b> Read-write. Reset: 0. Specifies the beginning address [31:12] of the Periodic Frame List in the system memory. If ( <code>EHCI[3:1]x08[AddressingCapability64Bit]==1</code> ), the most significant 32 bits of every control data structure address comes from <code>EHCI[3:1]x30</code> .
11:0	Reserved.

### EHCI[3:1]x38 Current Async List Address

Bits	Description
31:5	<b>LPL: Link Pointer Low.</b> Read-write. Reset: 0. Specifies the address [31:5] of the next asynchronous queue head to be executed. If ( <code>EHCI[3:1]x08[AddressingCapability64Bit]==1</code> ), the most significant 32 bits of every control data structure address comes from <code>EHCI[3:1]x30</code> . This field may only reference a Queue Head (QH).
4:0	Reserved.

**EHCI[3:1]x60 Configure Flag**

Bits	Description
31:1	Reserved.
0	<b>CF: Configure Flag.</b> Read-write. Reset: 0. This bit controls the default port-routing control logic. Software sets this bit as the last action in its process of configuring the host controller. 0=Port routing control logic default-routes each port to an implementation dependent classic host controller. 1=Port routing control logic default-routes all ports to this host controller.

**EHCI[3:1]x[70,6C,68,64] Port Status Control [4:1]**

Bits	Description																				
31:25	<b>DeviceAddress: Device Address.</b> Read-write. Reset: 0. Specifies the 7-bit USB device address for the device attached to and immediately downstream of the associated root port. 0=Indicates no device is present or support for this feature is not present.																				
24:23	<b>SuspendStatus: Suspend Status.</b> Read-only; updated-by-hardware. Reset: 0. These two bits are used by software to determine whether the most recent L1 suspend request was successful, specifically: <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Success: State transition was successful (ACK)</td> </tr> <tr> <td>01b</td> <td>Not Yet: Device was unable to enter the L1 state at this time (NYET)</td> </tr> <tr> <td>10b</td> <td>Not Supported: Device does not support the L1 state (STALL)</td> </tr> <tr> <td>11b</td> <td>Timeout/Error: Device failed to respond or an error occurred.</td> </tr> </tbody> </table> <p>This field is updated by hardware immediately following the completion of an L1 transition request (via an LPM Token). To avoid any race conditions with hardware, software should only consume the contents of this field when Suspend==0 (port no longer in L1).</p>	Bits	Definition	00b	Success: State transition was successful (ACK)	01b	Not Yet: Device was unable to enter the L1 state at this time (NYET)	10b	Not Supported: Device does not support the L1 state (STALL)	11b	Timeout/Error: Device failed to respond or an error occurred.										
Bits	Definition																				
00b	Success: State transition was successful (ACK)																				
01b	Not Yet: Device was unable to enter the L1 state at this time (NYET)																				
10b	Not Supported: Device does not support the L1 state (STALL)																				
11b	Timeout/Error: Device failed to respond or an error occurred.																				
22	<b>WakeOnOverCurrentEnable: wake on over-current enable.</b> Read-write. Reset: 0. 1=Enables the port to be sensitive to over-current conditions as wake-up events. This field is 0 if (PortPower==0).																				
21	<b>WakeOnDisconnectEnable: wake on disconnect enable.</b> Read-write. Reset: 0. 1=Enables the port to be sensitive to device disconnects as wake-up events. This field is 0 if (PortPower==0).																				
20	<b>WakeOnConnectEnable: wake on connect enable.</b> Read-write. Reset: 0. 1=Enables the port to be sensitive to device connects as wake-up events. This field is 0 if (PortPower==0).																				
19:16	<b>PortTestControl: Port Test Control.</b> Read-write. Reset: 0. When this field is 0, the port is not operating in a test mode. A non-zero value indicates that it is operating in test mode and the specific test mode is indicated by the specific value. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Bits</th> <th>TestMode</th> <th>Bits</th> <th>TestMode</th> </tr> </thead> <tbody> <tr> <td>0000b</td> <td>Test mode not enabled</td> <td>0100b</td> <td>Test Packet</td> </tr> <tr> <td>0001b</td> <td>Test J_STATE</td> <td>0101b</td> <td>Test FORCE_ENABLE</td> </tr> <tr> <td>0010b</td> <td>Test K_STATE</td> <td>1111-0110b</td> <td>Reserved</td> </tr> <tr> <td>0011b</td> <td>Test SE0_NAK</td> <td></td> <td></td> </tr> </tbody> </table>	Bits	TestMode	Bits	TestMode	0000b	Test mode not enabled	0100b	Test Packet	0001b	Test J_STATE	0101b	Test FORCE_ENABLE	0010b	Test K_STATE	1111-0110b	Reserved	0011b	Test SE0_NAK		
Bits	TestMode	Bits	TestMode																		
0000b	Test mode not enabled	0100b	Test Packet																		
0001b	Test J_STATE	0101b	Test FORCE_ENABLE																		
0010b	Test K_STATE	1111-0110b	Reserved																		
0011b	Test SE0_NAK																				
15:14	<b>PortIndicatorControl: Port Indicator Control.</b> IF (EHCI[3:1]x04[PortIndicators]==0) THEN Read-only. ELSE Read-write. ENDIF. Reset: 0. This field is 0 if PortPower==0. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Bits</th> <th>Definition</th> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Port indicators are off.</td> <td>10b</td> <td>Green</td> </tr> <tr> <td>01b</td> <td>Amber</td> <td>11b</td> <td>Undefined</td> </tr> </tbody> </table>	Bits	Definition	Bits	Definition	00b	Port indicators are off.	10b	Green	01b	Amber	11b	Undefined								
Bits	Definition	Bits	Definition																		
00b	Port indicators are off.	10b	Green																		
01b	Amber	11b	Undefined																		

13	<p><b>PortOwner: Port Owner.</b> Read-write. Reset: 1. 1=A companion host controller owns and controls the port. Software writes a 1 to this bit when the attached device is not a high-speed device. This bit unconditionally goes to a 0 when <code>EHCI[3:1]x60[CF]</code> transitions from 0 to 1. This bit unconditionally goes to 1b whenever <code>EHCI[3:1]x60[CF]==0</code>. System software uses this field to release ownership of the port to a selected host controller (in the event that the attached device is not a high-speed device).</p>										
12	<p><b>PortPower: Port Power.</b> IF (<code>EHCI[3:1]x04[PPC]==0</code>) THEN Read-only. ELSE Read-write. ENDIF. IF (<code>EHCI[3:1]x04[PPC]==0</code>) THEN Reset: 1. ELSE Reset: X. ENDIF. 0=Power is off; the port is non-functional and will not report attaches, detaches, etc. 1=Power is on. When an over-current condition is detected on a powered port and <code>EHCI[3:1]x04[PPC]==1</code>, the PortPower bit in each affected port may be transitioned by the host controller from 1 to 0 (removing power from the port).</p>										
11:10	<p><b>LineStatus[1:0]: Line Status.</b> Read-only. Reset: X. These bits reflect the current logical levels of the D+ (LineStatus[1]) and D- (LineStatus[0]) signal lines. These bits are used for detection of low-speed USB devices prior to the port reset and enable sequence. This field is valid only when <code>PortEnable==0</code> and <code>CurrentConnectStatus==1</code>.</p> <table border="1" data-bbox="264 699 1450 867"> <thead> <tr> <th data-bbox="264 699 508 730"><u>Bits</u></th> <th data-bbox="516 699 1450 730"><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td data-bbox="264 730 508 762">00b</td> <td data-bbox="516 730 1450 762">SE0. Not Low-speed device, perform EHCI reset</td> </tr> <tr> <td data-bbox="264 762 508 793">01b</td> <td data-bbox="516 762 1450 793">K-state. Low-speed device, release ownership of port</td> </tr> <tr> <td data-bbox="264 793 508 825">10b</td> <td data-bbox="516 793 1450 825">J-state. Not Low-speed device, perform EHCI reset</td> </tr> <tr> <td data-bbox="264 825 508 867">11b</td> <td data-bbox="516 825 1450 867">Undefined. Not Low-speed device, perform EHCI reset.</td> </tr> </tbody> </table> <p>This value of this field is undefined if (<code>PortPower==0</code>).</p>	<u>Bits</u>	<u>Definition</u>	00b	SE0. Not Low-speed device, perform EHCI reset	01b	K-state. Low-speed device, release ownership of port	10b	J-state. Not Low-speed device, perform EHCI reset	11b	Undefined. Not Low-speed device, perform EHCI reset.
<u>Bits</u>	<u>Definition</u>										
00b	SE0. Not Low-speed device, perform EHCI reset										
01b	K-state. Low-speed device, release ownership of port										
10b	J-state. Not Low-speed device, perform EHCI reset										
11b	Undefined. Not Low-speed device, perform EHCI reset.										
9	<p><b>SuspendUsingL1: Suspend using L1.</b> Read-write. Reset: 0. 0=The host controller will employ the legacy (L2) suspend mechanism. 1=If a non-zero value is specified in DeviceAddress, the host controller will generate an LPM Token to enter the L1 state whenever software writes a 1 to the Suspend bit, as well as L1 exit timing during any device- or host-initiated resume. Software should only set this bit when the device attached immediately downstream of this root port supports L1 transitions.</p>										
8	<p><b>PortReset: Port Reset.</b> Read-write. Reset: 0. 1=Port is in Reset. 0=Port is not in Reset. Software writes 1 to start the bus reset sequence; It must keep this bit at a 1 long enough to ensure the reset sequence completes. Software writes 0 to terminate the bus reset sequence; This bit will not read as a 0 until after the reset has completed.</p> <p>When software writes this bit to a 1, it must also write a 0 to the PortEnable bit. If the port is in high-speed mode after reset is complete, the host controller will automatically enable this port (e.g. set the Port Enable bit to a 1). <code>EHCI[3:1]x24[HCHalted]</code> should be a 0 before software attempts to use this bit. The host controller may hold Port Reset asserted to a 1 when the HCHalted bit is a 1. This field is 0 if <code>PortPower==0</code>.</p>										
7	<p><b>Suspend.</b> Read; cleared-by-hardware; write-1-only. Reset: 0. 0=IF (<code>EHCI[3:1]x[70,6C,68,64][PortEnable] == 1</code>) THEN Port is enabled. ELSE Port is disabled. ENDIF. 1=IF (<code>EHCI[3:1]x[70,6C,68,64][PortEnable] == 1</code>) THEN Port is in suspend state; The port can be in either L1 or L2 suspend state depending on SuspendUsingL1. ELSE Port is disabled. ENDIF.</p> <p>When in suspend state, downstream propagation of data is blocked on this port, except for port reset. If this bit is written with a 1 when a transaction is in progress then the blocking will not occur until the end of the current transaction. In the suspend state, the port is sensitive to resume detection. Additional status for L1-based transitions is provided to software via the SuspendStatus field. This field is 0 if <code>PortPower==0</code>. If system software sets this bit to a 1 when the port is not enabled (i.e. <code>PortEnable == 0</code>) the results are undefined. A write of 0 to this bit is ignored by the host controller. The host controller will unconditionally set this bit to a 0 when:</p> <ul style="list-style-type: none"> <li>• Software sets the ForcePortResume bit to a 0 from a 1.</li> <li>• Software sets the PortReset bit to a 1 from a 0.</li> <li>• Whenever PortPower is zero.</li> </ul>										

6	<p><b>ForcePortResume: Force Port Resume.</b> Read-write; set-by-hardware. Reset: 0. 1=Resume detected/driven on port. 0=No resume (K-state) detected/driven on port. This field is 0 if PortPower==0.</p> <p>This functionality defined for manipulating this bit depends on the value of the Suspend and SuspendUsingL1 bits. For example, if the port is not suspended (Suspend and Enabled bits are a 1) and software transitions this bit to a 1, then the effects on the bus are undefined.</p> <p>The Host Controller sets this bit to a 1 if a J-to-K transition is detected while the port is in the Suspend state. In this case, EHCI[3:1]x24[PortChangeDetect] and/or EHCI[3:1]x24[PortNChangeDetect] will also be set to 1.</p> <p>Software sets this bit to a 1 to drive resume signaling. If software sets this bit to a 1, the host controller must not set EHCI[3:1]x24[PortChangeDetect] and/or EHCI[3:1]x24[PortNChangeDetect]. Writing a 0 (from 1) causes the port to return to high-speed mode. This bit will remain a 1 until the port has switched to the high-speed idle. The host controller must complete this transition within 2 milliseconds of software setting this bit to a 0.</p> <p>For legacy (L2) transitions, software must appropriately time the Resume and set this bit to a 0 when the appropriate amount of time has elapsed. Software does not need to time resume signaling for L1 transactions as host controller hardware will automatically enforce the necessary timing and clear this bit when the port has fully resumed. Software can influence the amount of time hardware will drive resume signaling during L1 exit via EHCI[3:1]x20[HostInitiatedResumeDuration].</p>
5	<p><b>OverCurrentChange: over-current change.</b> Read; set-by-hardware; write-1-to-clear. Reset: 0. 1=There is a change to OverCurrentActive.</p>
4	<p><b>OverCurrentActive: over-current active.</b> Read-only; updated-by-hardware. Reset: 0. 1=This port currently has an over-current condition. 0=This port does not have an over-current condition.</p>
3	<p><b>PortEnableDisableChange: Port Enable/Disable Change.</b> Read; set-by-hardware; write-1-to-clear. Reset: 0. 1=Port enabled/disabled status has changed. 0=No change. For the root hub, this bit gets set to a 1 only when a port is disabled due to the appropriate conditions existing at the EOF2. This field is 0 if PortPower==0.</p>
2	<p><b>PortEnable: port enable.</b> Read-write; set-by-hardware. Reset: 0. 1=Enable. 0=Disable.</p> <p>Ports can only be enabled by the host controller as a part of the reset and enable. Software cannot enable a port by writing a 1 to this field. The host controller will only set this bit to a 1 when the reset sequence determines that the attached device is a high-speed device.</p> <p>Ports can be disabled by either a fault condition (disconnect event or other fault condition) or by host software. The bit status does not change until the port state actually changes. There may be a delay in disabling or enabling a port due to other host controller and bus events. When the port is disabled, downstream propagation of data is blocked on this port, except for reset. This field is 0 if PortPower==0.</p>
1	<p><b>ConnectStatusChange: Connect Status Change.</b> Read; set-by-hardware; write-1-to-clear. Reset: 0. Indicates a change has occurred in the port's CurrentConnectStatus. The host controller sets this bit for all changes to the port device connect status, even if system software has not cleared an existing connect status change. This field is 0 if PortPower==0. 1=Change in Current Connect Status. 0=No change.</p>
0	<p><b>CurrentConnectStatus: Current Connect Status.</b> Read-only. Reset: 0. This value reflects the current state of the port, and may not correspond directly to the event that caused the ConnectStatusChange to be set to 1. This field is 0 if PortPower==0. 1=Device is present on port. 0=No device is present.</p>

#### **EHCI[3:1]xA4 Packet Buffer Threshold Values**

BIOS: 0040\_0040h.



Bits	Description
31:24	Reserved.
23:16	<b>OUTThreshold: OUT Threshold.</b> Read-write. Reset: 40h. The transmit packet starts at UTMI interface when threshold of internal FIFO for transmit packet is reached. The value represents multiple of 8 bytes. For example, 10h means 128 bytes. The smallest acceptable value is 08h (64 bytes).
15:8	Reserved.
7:0	<b>INThreshold: IN Threshold.</b> Read-write. Reset: 40h. The PCI transaction starts when threshold of internal FIFO for receive packet is reached. The value represents multiple of 8 bytes. For example, 10h means 128 bytes. The smallest acceptable value is 08h (64 bytes).

**EHCI[3:1]xB4 UTMI Control**

BIOS: See 2.15.2.6&2.15.2.8.

Bits	Description																												
31:18	Reserved.																												
17	<b>VBusy.</b> Read-only. Reset: 0. 1=Block software writes to bits[16:8] when port router is updating the field.																												
16:13	<b>PortNumber: Port Number.</b> Read-write. Reset: 0. Select the corresponding port PHY or common block to load the VControl bits. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Bits</th> <th>Definition</th> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>0000b</td> <td>Port 0</td> <td>0011b</td> <td>Port 3</td> </tr> <tr> <td>0001b</td> <td>Port 1</td> <td>1110b-0100b</td> <td>Reserved</td> </tr> <tr> <td>0010b</td> <td>Port 2</td> <td>1111b</td> <td>Common block</td> </tr> </tbody> </table>	Bits	Definition	Bits	Definition	0000b	Port 0	0011b	Port 3	0001b	Port 1	1110b-0100b	Reserved	0010b	Port 2	1111b	Common block												
Bits	Definition	Bits	Definition																										
0000b	Port 0	0011b	Port 3																										
0001b	Port 1	1110b-0100b	Reserved																										
0010b	Port 2	1111b	Common block																										
12	<b>VLoadB.</b> Read-write. Reset: 1. VControl value load, active low. 0=Load the new VControl value to the PHY or common block specified by PortNumber. VLoadB must be 1 while VControlModeSel is changed.																												
11	Reserved.																												
10:7	<b>VControlModeSel.</b> Read-write. Reset: 0. Selects the PHY control mode group. There are 8 groups defined for PHY control mode. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Bits</th> <th>Definition</th> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>0000b</td> <td>Group 0</td> <td>0110b</td> <td>Group 6. Analog Control</td> </tr> <tr> <td>0001b</td> <td>Group 1</td> <td>0111b</td> <td>Group 7</td> </tr> <tr> <td>0010b</td> <td>Group 2</td> <td>1000b</td> <td>Group 8</td> </tr> <tr> <td>0011b</td> <td>Group 3</td> <td>1001b</td> <td>Group 9</td> </tr> <tr> <td>0100b</td> <td>Group 4. CDR DFT control</td> <td>1111b-1010b</td> <td>Reserved</td> </tr> <tr> <td>0101b</td> <td>Group 5. CDR DFT control</td> <td></td> <td></td> </tr> </tbody> </table>	Bits	Definition	Bits	Definition	0000b	Group 0	0110b	Group 6. Analog Control	0001b	Group 1	0111b	Group 7	0010b	Group 2	1000b	Group 8	0011b	Group 3	1001b	Group 9	0100b	Group 4. CDR DFT control	1111b-1010b	Reserved	0101b	Group 5. CDR DFT control		
Bits	Definition	Bits	Definition																										
0000b	Group 0	0110b	Group 6. Analog Control																										
0001b	Group 1	0111b	Group 7																										
0010b	Group 2	1000b	Group 8																										
0011b	Group 3	1001b	Group 9																										
0100b	Group 4. CDR DFT control	1111b-1010b	Reserved																										
0101b	Group 5. CDR DFT control																												
6:0	Reserved.																												

**EHCI[3:1]xB8 Loopback Test**

Bits	Description
31:27	Reserved.

26:21	<p><b>StatusOfPortsforPowerUpStateCheck[5:0]: Status of ports for PowerUp State Check.</b> Read-only. Reset: 0. Status bit to indicate the Power Up state auto-checking result from the individual port. Each bit maps to a corresponding port, from port 0 to port 5.</p> <table border="1"> <thead> <tr> <th>Bit</th> <th>Definition</th> <th>Bit</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>[0]</td> <td>Port 0</td> <td>[3]</td> <td>Port 3</td> </tr> <tr> <td>[1]</td> <td>Port 1</td> <td>[4]</td> <td>Port 4</td> </tr> <tr> <td>[2]</td> <td>Port 2</td> <td>[5]</td> <td>Port 5</td> </tr> </tbody> </table> <p>For each bit, 0=PHY Power Up State checking is fail. 1=PHY Power Up State checking is good.</p>	Bit	Definition	Bit	Definition	[0]	Port 0	[3]	Port 3	[1]	Port 1	[4]	Port 4	[2]	Port 2	[5]	Port 5
Bit	Definition	Bit	Definition														
[0]	Port 0	[3]	Port 3														
[1]	Port 1	[4]	Port 4														
[2]	Port 2	[5]	Port 5														
20	<p><b>EnablePHYPowerUpStateChecking: Enable PHY PowerUp State Checking.</b> Read-write. Reset: 0. 1=Enable auto-checking on PHY Power Up State. There is built-in logic to check the PHY default Power Up state to detect manufacturing defects in the PHY macro.</p>																
19:12	<p><b>GoodReceivedPacketCount: Good Received Packet Count.</b> Read-only; updated-by-hardware. Reset: 0. Specified the number of good packets that the host controller received during the loopback test mode. These bits will be cleared by clearing EnableLoopBackTest.</p>																
11	Reserved.																
10	<p><b>LoopbackTestDone: Loopback Test Done.</b> Read-only. Reset: 0. 1=Loopback test is done.</p>																
9	<p><b>LoopbackTestStatus: Loopback Test Status.</b> Read-only. Reset: 0. 0=CRC error on loopback receiving data. 1=Good CRC on loopback receiving data.</p>																
8	<p><b>EnableLoopBackTest: Enable Loop Back Test.</b> Read-write. Reset: 0. 1=Enable external USB port loopback test. The loopback test is to set one port to TX mode (Test Packet mode) and one port to RX mode (Test SEO_NAK). See <a href="#">EHCI[3:1]x[70,6C,68,64][PortTestControl]</a> for information on the tests.</p>																
7:4	Reserved.																
3:0	<p><b>ReceivedPacketCount: Received Packet Count.</b> Read-write. Reset: 0. RX data packet count. This counter defines the number (in power of 16) of RX data packet that should be checked for the loop back test.</p>																

### **EHCI[3:1]xBC EOR MISC Control**

Bits	Description
31:24	<p><b>ForceTxData[7:0]: Force TX Data [7:0].</b> Read-write. Reset: 0. Used to force TxData[7:0] when the port is in TEST_K mode. This can be used to force PHY to generate a desired output pattern for PHY debugging and characterization purposes.</p>
23:15	Reserved.
14	<p><b>EHCIDeepBlinkPowerSavingEnable: EHCI Deep Blink Power Saving Enable.</b> Read-write. Reset: 1. 1=Enable Deep Blink power saving clock gating; EHCI will request to stop the Global Blink clock when the list processor is idle.</p>
13	Reserved.
12	<p><b>EHCIPowerSavingEnable: EHCI Power Saving Enable.</b> Read-write. Reset: 1. 1=Enable power saving clock gating. When enabled, dynamic clock gating is enabled when EHCI is not in operational mode. The clock going to all memory modules will be gated off. Blink clock also gets gated off unless the connection interrupt is detected.</p>
11	<p><b>DisablesHSuFrameBabbleDetection: Disables HS uFrame Babble detection.</b> Read-write. Reset: 0. 1=Disable HS uFrame babble detection.</p>
10:8	Reserved.

7:4	<b>InterPacketGapAdjustCounter: Inter-packet Gap Adjust Counter.</b> Read-write. Reset: 4h. Specifies the counter used to adjust the inter-packet gap for test packet.
3:2	Reserved.
1	<b>InterruptrouteControlonForcePortResume: Interrupt route Control on ForcePortResume.</b> Read-write. Reset: 0. This bit only takes effect when the EnableInterruptOnForcePortResume=1. 0=Report interrupt to <a href="#">EHCI[3:1]x24[UsbInt]</a> bit on software clear <a href="#">EHCI[3:1]x[70,6C,68,64][ForcePortResume]</a> . 1=Report interrupt to <a href="#">EHCI[3:1]x24[PortChangeDetect]</a> bit on software clear <a href="#">EHCI[3:1]x[70,6C,68,64][ForcePortResume]</a> .
0	<b>EnableInterruptonForcePortResume: Enable Interrupt on ForcePortResume.</b> Read-write. Reset: 0. 1=Enable host controller to generate interrupt on software clear <a href="#">EHCI[3:1]x[70,6C,68,64][ForcePortResume]</a> .

### **EHCI[3:1]xC0 USB Common PHY CAL and Control**

Bits	Description
31	Reserved.
30:24	<b>CommonPhyCalBus.</b> Read-only; updated-by-hardware. Reset: X. PHY Common Calibration Bus.
23	Reserved.
22:20	<b>ComCalAmp.</b> Read-only; updated-by-hardware. Reset: X. Specifies the amplitude calibration data from PHY.
19:18	Reserved.
17	<b>AddToCommonCalibration.</b> Read-write. Reset: 1. BIOS: See <a href="#">2.15.2.7</a> . 1=The signed NewCalBus is added to the ComCalBus and returned to the PHY ports; Any overflow is clamped to all 1s; Any underflow is clamped to all 0s. 0=The signed NewCalBus replaces the ComCalBus and returns to the PHY ports if UseCommonCalibration=0.
16	<b>UseCommonCalibration.</b> Read-write. Reset: 0. BIOS: See <a href="#">2.15.2.7</a> . 1=The PHY's calibration value in ComCalBus is returned to the PHY ports. 0=The value after adjustment is returned to the PHY ports.
15:8	<b>NewCalBus.</b> Read-write. Reset: 0. BIOS: See <a href="#">2.15.2.7</a> . New calibration bus signed value.
7	Reserved.
6:0	<b>ComCalBus.</b> Read-only; updated-by-hardware. Reset: X. Calibration bus value from PHY before adjustment.

### **EHCI[3:1]xC4 USB Common PHY Control 1**

BIOS: See [2.15.2.6](#).

Bits	Description										
31:24	<b>DLLControl: DLL Control.</b> Read-write. Reset: A0h. <table border="0"> <tr> <td><u>Bit</u></td> <td><u>Definition</u></td> </tr> <tr> <td>[1:0]</td> <td>Reserved.</td> </tr> <tr> <td>[2]</td> <td>DLL_EN_PFDphases. 1=Enable DLL phase-sampling based lock detection.</td> </tr> <tr> <td>[5:3]</td> <td>DLL_cpump. DLL charge pump current control.</td> </tr> <tr> <td>[7:6]</td> <td>DLL_VtoI. DLL gain control.</td> </tr> </table>	<u>Bit</u>	<u>Definition</u>	[1:0]	Reserved.	[2]	DLL_EN_PFDphases. 1=Enable DLL phase-sampling based lock detection.	[5:3]	DLL_cpump. DLL charge pump current control.	[7:6]	DLL_VtoI. DLL gain control.
<u>Bit</u>	<u>Definition</u>										
[1:0]	Reserved.										
[2]	DLL_EN_PFDphases. 1=Enable DLL phase-sampling based lock detection.										
[5:3]	DLL_cpump. DLL charge pump current control.										
[7:6]	DLL_VtoI. DLL gain control.										
23:21	Reserved.										

20	<b>PLLBypass: PLL Bypass.</b> Read-write. Reset: 0. 1=Enable USB Common PLL bypass.
19:16	<b>DutyAdj.</b> Read-write. Reset: 4h. Specifies CLK480 duty cycle control from 40-60% to 60-40%.
15:12	<b>PVI.</b> Read-write. Reset: 2h. Specifies PLL V-I Converter Control for common block PLL.
11:8	<b>IRefAdj.</b> Read-write. Reset: 1. Specifies internal reference bias adjustment for common block.
7:4	<b>XRefAdj.</b> Read-write. Reset: 1. Specifies external reference bias adjustment for common block.
3:0	<b>CPAdj.</b> Read-write. Reset: 4h. Charge pump setting for common block PLL.

### **EHCI[3:1]xD0 USB Common PHY Control 2**

Bits	Description
31:16	Reserved.
15	<b>BgBypass.</b> Read-write. Reset: 0. Bandgap bypass.
14	Reserved.
13:12	<b>PIILockAdj[1:0].</b> Read-write. Reset: 0. PLL lock detection tuning.
11	<b>UnlockSticky.</b> Read-only. Reset: X. Sticky unlock detection observability.
10	<b>UnlockReset.</b> Read-write. Reset: 0. Reset sticky un-lock detector.
9	<b>LockDetect.</b> Read-only. Reset: X. Lock detection observability.
8	<b>EnLockDetect.</b> Read-write. Reset: 0. Enable PLL lock detection.
7:4	<b>CBackup.</b> Read-write. Reset: 0h. Common block backup.
3:0	<b>BgAdj.</b> Read-write. Reset: 6. BIOS: See <a href="#">2.15.2.6</a> . Bandgap voltage adjust.

### **EHCI[3:1]xD4 USB Common PHY Control 3**

BIOS: See [2.15.2.6](#).

Bits	Description
31:8	Reserved.
7:6	<b>PIIFilter.</b> Read-write. Reset: 0. PLL Filter tuning setting.
5:4	<b>PIIDiv.</b> Read-write. Reset: 0. PLL divider setting.
3:2	<b>PIIReg.</b> Read-write. Reset: 0. PLL Regulator tuning setting.
1	<b>CalEnable.</b> Read-write. Reset: 0. Calibration Enable.
0	<b>BgStart.</b> Read-write. Reset: 0. Bandgap startup tuning setting.

### **EHCI[3:1]xDC USB Battery Charger Enable**

Bits	Description
31:4	Reserved.
3:0	<b>UsbBatteryChargerEnable.</b> Read-write. Reset: 0. One bit controls one respective port. For each bit, 1=Enable USB PHY battery charger function for the port.

**3.25.4.3 USB 3.0 (xHCI)**

The xHCI controller has a PCI configuration register space. The PCI configuration register space uses device 10h function 0 and is defined below as D10F0xXX.

**3.25.4.3.1 Device 10h Function 0 (xHCI) 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.

**D10F0x00 Device/Vendor ID**

Bits	Description
31:16	<b>DeviceID: Device ID.</b> Read-only. Reset: 7814h. Device Identifier.
15:0	<b>VendorID: Vendor ID.</b> Read-only. Reset: 1022h. Vendor Identifier.

**D10F0x04 Status/Command**

Bits	Description
31	<b>DetectedParityError: Detected Parity Error.</b> Read; set-by-hardware; write-1-to-clear. Reset: 0.
30	<b>SignaledSystemError: Signaled System Error.</b> Read; set-by-hardware; write-1-to-clear. Reset: 0. 1=System error signaled via SERR#.
29	<b>ReceivedMasterAbort: Received Master Abort.</b> Read; set-by-hardware; write-1-to-clear. Reset: 0.
28	<b>ReceivedTargetAbort: Received Target Abort.</b> Read; set-by-hardware; write-1-to-clear. Reset: 0.
27	<b>SignaledTargetAbort: Signaled Target Abort.</b> Read; set-by-hardware; write-1-to-clear. Reset: 0.
26:25	<b>DEVSELTiming: DEVSEL Timing.</b> Value: 0.
24	<b>MasterDataParityError: Master Data Parity Error.</b> Read; set-by-hardware; write-1-to-clear. Reset: 0.
23	<b>FastBack2BackCapable: Fast Back-to-Back Capable.</b> Value: 0.
22	Reserved.
21	<b>Capable66MHz: 66 MHz Capable.</b> Value: 0.
20	<b>CapabilitiesList: Capabilities List.</b> Value: 1. 1=Indicates that the value read at offset 34h is a pointer in Configuration Space to a linked list of new capabilities.
19	<b>InterruptStatus: Interrupt Status.</b> Read-only. Reset: 0. 1=An INTx interrupt Message is pending in the device.
18:11	Reserved.
10	<b>InterruptDisable: Interrupt Disable.</b> Read-only. Reset: 0. 0=Enable the assertion of the device/function's INTx# signal. 1=Disable the assertion of the device/function's INTx# signal.
9	<b>FastBack2BackEnable: Fast Back-to-Back Enable.</b> Value: 0. 0=Only fast back-to-back transactions to the same agent are allowed.
8	<b>SERREnable: SERR# Enable.</b> Read-write. Reset: 0. 1=System error reporting enabled.
7	RAZ.

6	<b>ParityEnable: Parity Enable.</b> Read-write. Reset: 0. 0=The device sets its DetectedParityError when an error is detected, but continues normal operations without asserting PERR#. 1=The device must take its normal action when a parity error is detected.
5	<b>VgaPaletteRegisterAccesses: VGA palette register accesses.</b> Value: 0. 0=The device should treat palette write accesses like all other accesses.
4	<b>MemoryWriteandInvalidateCommand: Memory Write and Invalidate Command.</b> Value: 0. 0=Memory Write must be used.
3	<b>SpecialCycle: Special Cycle.</b> Value: 0. 0=No Special Cycle support.
2	<b>BusMaster: Bus Master.</b> Read-write. Reset: 0. 0=Disable the device from generating PCI accesses. 1=Allow the device to behave as a bus master.
1	<b>MemorySpaceAccesses: Memory Space Accesses.</b> Read-write. Reset: 0. 0=Disable the device response. 1=Allow the device to respond to Memory Space accesses.
0	<b>IOSpaceAccesses: IO Space Accesses.</b> Value: 0. 0=IO accesses targeting this device are not accepted.

**D10F0x08 Revision ID / Class Code**

Bits	Description
31:24	<b>BaseClass: Base Class.</b> Read-only. Reset: Ch. Base Class. Serial Bus Controller.
23:16	<b>SubClass: Sub Class.</b> Read-only. Reset: 3h. Sub Class. Universal Serial Bus Host Controller.
15:8	<b>ProgrammingInterface: Programming Interface.</b> Read-only. Reset: 30h. Programming Interface. USB 3.0 Host Controller that conforms to xHC specification.
7:0	<b>RevisionID.</b> Read-only. Reset: 01h.

**D10F0x0C Miscellaneous**

Bits	Description
31:24	<b>Bist.</b> Value: 0. 00h=BIST is not supported.
23:16	<b>HeaderType: Header Type.</b> Value: 00h. 00h=The device does not have multiple functions.
15:8	<b>LatencyTimer: Latency Timer.</b> Value: 0.
7:0	<b>CacheLineSize: Cache Line Size.</b> Read-write. Reset: 0. This read/write field specifies the system cache line size in units of doublewords and must be initialized to 00h.

**D10F0x10 Bar 0**

Bits	Description
31:13	<b>Bar0.</b> Read-write. Reset: 0. Specifies Base Address [31:4].
12:4	Reserved.
3	<b>PM: Prefetch memory.</b> Value: 0. 0=Indicates that there is no support for prefetchable memory.
2:1	<b>Tp: Type.</b> Value: 10b. 10b=64 bit address.
0	<b>Ind: Indicator.</b> Value: 0. 0=Indicates that the operational registers of the device are mapped into memory space of the main memory of the PC host system.

**D10F0x14 Bar 1**

Bits	Description
31:0	<b>Bar1: Base Address High.</b> Read-write. Reset: 0. Specifies Base Address [63:32].

**D10F0x2C Subsystem Vendor ID / Subsystem ID**

Bits	Description
31:16	<b>SubsystemID: Subsystem ID.</b> Read-only. Reset: 7814h. Subsystem ID.
15:0	<b>SubsystemVendorID: Subsystem Vendor ID.</b> Read-only. Reset: 1022h. Subsystem Vendor ID.

**D10F0x34 Capability Pointer**

Bits	Description
7:0	<b>CapabilityPointer: Capability Pointer.</b> Read-only. Reset: 50h. Address of the 1st element of capability link.

**D10F0x3C Interrupt Line**

Bits	Description
31:16	Reserved.
15:8	<b>InterruptPin: Interrupt Pin.</b> Read-only. Reset: 1. Specifies which interrupt pin the device (or device function) uses. A value of 1 corresponds to INTA#.
7:0	<b>InterruptLine: Interrupt Line.</b> Read-write. Reset: 0. Specifies which input of the system interrupt controller(s) the device's interrupt pin is connected to. The device itself does not use this value; rather it is used by device drivers and operating systems to determine priority and vector information.

**D10F0x40 IDP Index Register**

Bits	Description
31:13	Reserved.
12:2	<b>IDPIndex: IDP Index.</b> Read-only. Reset: 0. This register selects the doubleword offset of the memory mapped register to be accessed. All register accesses are at doubleword granularity.
1:0	Reserved.

**D10F0x44 IDP Data Register**

Bits	Description
31:0	<b>IDPData: IDP Data.</b> Read-only. Reset: 0. Specifies the data read from or written to the memory mapped register pointed to by <a href="#">D10F0x40</a> . A physical register is not actually implemented as the data is actually stored in the memory mapped registers. The default value is the same as the default value of the register pointed to by <a href="#">D10F0x40</a> .

**D10F0x48 Indirect PCI Index Register**

Bits	Description												
31:30	<b>IndirectPCIIndex[31:30]: Indirect PCI Index.</b> Read-write. Reset: 0. Selects the PCI indirect space. This field is combined with IndirectPCIIndex[15:2] to select the PCI indirect register to be accessed. <ul style="list-style-type: none"> <li>IndirectPCIIndex[31:0] range for <a href="#">3.25.4.3.1.1 [USB 3.0 125 MHz PCI Indirect Space]</a>: 0000_FFFFh-0000_0000h.</li> <li>IndirectPCIIndex[31:0] range for <a href="#">3.25.4.3.1.2 [USB 3.0 60 MHz PCI Indirect Space]</a>: 4000_FFFFh-4000_0000h.</li> </ul> <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Selects 125 MHz PCI indirect space</td> <td>10b</td> <td>Reserved</td> </tr> <tr> <td>01b</td> <td>Selects 60 MHz PCI indirect space</td> <td>11b</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Definition	Bits	Definition	00b	Selects 125 MHz PCI indirect space	10b	Reserved	01b	Selects 60 MHz PCI indirect space	11b	Reserved
Bits	Definition	Bits	Definition										
00b	Selects 125 MHz PCI indirect space	10b	Reserved										
01b	Selects 60 MHz PCI indirect space	11b	Reserved										
29:16	Reserved.												
15:2	<b>IndirectPCIIndex[15:2]: Indirect PCI Index.</b> Read-write. Reset: 0. Selects the doubleword offset of the PCI indirect space register to be accessed. All register accesses are at doubleword granularity.												
1:0	Reserved.												

**D10F0x4C Indirect PCI Data Register**

Bits	Description
31:0	<b>IndirectPCIData: Indirect PCI Data.</b> Read-write. Specifies the data written to/read from the PCI indirect space register pointed to by <a href="#">D10F0x48</a> . A physical register is not actually implemented as the data is actually stored in the PCI indirect space registers. The default value is the same as the default value of the register pointed to by <a href="#">D10F0x48</a> .

**3.25.4.3.1.1 USB 3.0 125 MHz PCI Indirect Space**

These registers are indirectly accessed through [D10F0x48](#) and [D10F0x4C](#).

**D10F0x4C\_x08 Port Disable Write Once**

Bits	Description
------	-------------



31:2	Reserved.						
1:0	<p><b>XhcPortDisableWriteOnce: xHC Port Disable Write Once.</b> Read; write-once; write-1-only. Reset: 0. This field is bit significant. Once written, the register can only be cleared by PciRst#. For each bit, 1=Disable the corresponding port.</p> <table border="1"> <thead> <tr> <th><u>Bit</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>[0]</td> <td>port 0</td> </tr> <tr> <td>[1]</td> <td>port 1</td> </tr> </tbody> </table> <p>If (D10F0x4C_x08[XhcPortDisableWriteOnce]   D10F0x4C_x0C[XhcPortDisableRW]), then the corresponding xHC port is disabled.</p>	<u>Bit</u>	<u>Definition</u>	[0]	port 0	[1]	port 1
<u>Bit</u>	<u>Definition</u>						
[0]	port 0						
[1]	port 1						

**D10F0x4C\_x0C Port Disable RW**

Bits	Description						
31:2	Reserved.						
1:0	<p><b>XhcPortDisableRW: xHC Port Disable RW.</b> Read-write. Reset: 0. This field is bit significant. For each bit, 1=The corresponding port is disabled.</p> <table border="1"> <thead> <tr> <th><u>Bit</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>[0]</td> <td>port 0</td> </tr> <tr> <td>[1]</td> <td>port 1</td> </tr> </tbody> </table> <p>If (D10F0x4C_x08[XhcPortDisableWriteOnce]   D10F0x4C_x0C[XhcPortDisableRW]), then the corresponding xHC port is disabled.</p>	<u>Bit</u>	<u>Definition</u>	[0]	port 0	[1]	port 1
<u>Bit</u>	<u>Definition</u>						
[0]	port 0						
[1]	port 1						

**D10F0x4C\_x10 USB DCLK Event Counter 0**

Bits	Description
31:0	<b>UsbDclkEventCnt0Lo.</b> Read-only. Reset: 0. USB DClk Event Counter 0[31:0].

**D10F0x4C\_x14 USB DCLK Event Counter 1**

Bits	Description
31:0	<b>UsbDclkEventCnt1Lo.</b> Read-only. Reset: 0. USB DClk Event Counter 1[31:0].

**D10F0x4C\_x18 USB DCLK Event Counter Select**

Bits	Description
31:24	<b>UsbDclkEventCnt1Hi.</b> Read-only. Reset: 0. USB DClk Event Counter 1[39:32].
23:16	<b>UsbDclkEventCnt0Hi.</b> Read-only. Reset: 0. USB DClk Event Counter 0[39:32].
15:8	<b>UsbDclkEventSel1.</b> Read-write. Reset: 0. Select the event to be counted by Usb DClk Event Counter 1.
7:0	<b>UsbDclkEventSel0.</b> Read-write. Reset: 0. Select the event to be counted by Usb DClk Event Counter 0.

**D10F0x4C\_x1C USB DCLK Event Counter Control**

Bits	Description
31:3	Reserved.
2	<b>UsbDclkEventCntShadow</b> . Read-write. Reset: 0. Transfer USB DCLK event counter to shadow register.
1	<b>UsbDclkEventCntReseth</b> . Read-write. Reset: 1. Active low reset for USB DCLK event counters.
0	<b>UsbDclkEventCntEn</b> . Read-write. Reset: 0. 1=Enable USB DCLK event counters.

**3.25.4.3.1.2 USB 3.0 60 MHz PCI Indirect Space**

These registers are accessed through [D10F0x48](#) and [D10F0x4C](#).

**D10F0x4C\_x4000\_0000 UTMI Control**

BIOS: See [2.15.2.9](#), [2.15.2.11](#) & [2.15.2.12](#).

Bits	Description												
31:18	Reserved.												
17	<b>VBusy</b> . Read-only. Reset: 0.												
16:13	<b>PortNumber: Port Number</b> . Read-write. Reset: 0. Selects the corresponding port PHY or common block to load the VControl bits. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Bits</th> <th>Definition</th> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>0000b</td> <td>Port 0</td> <td>1110b-0010b</td> <td>Reserved</td> </tr> <tr> <td>0001b</td> <td>Port 1</td> <td>1111b</td> <td>Common block</td> </tr> </tbody> </table>	Bits	Definition	Bits	Definition	0000b	Port 0	1110b-0010b	Reserved	0001b	Port 1	1111b	Common block
Bits	Definition	Bits	Definition										
0000b	Port 0	1110b-0010b	Reserved										
0001b	Port 1	1111b	Common block										
12	<b>VLoadB</b> . Read-write. Reset: 1. Update PHY control mode (active low). 0=Load the new VControl value to PHY/common block. 1=Only load VControlModeSel value to PHY to select different PHY status group (see <a href="#">D10F0x4C_x4000_0004</a> ); VControl[6:0] value inside PHY won't get affected.												
11	Reserved.												
10:7	<b>VControlModeSel</b> . Read-write. Reset: 0. Selects the PHY control mode group. See: <a href="#">EHCI[3:1]xB4[VControlModeSel]</a> .												
6:0	<b>VControl</b> . Read-write. Reset: 24h. Controls PHY setting.												

**D10F0x4C\_x4000\_0004 USB PHY Status**

Bits	Description
31:16	Reserved.
15:8	<b>Port1PhyStatus</b> . Read-only; updated-by-hardware. Reset: 21h. Specifies the PHY Status of Port 1. Use the <a href="#">D10F0x4C_x4000_0000[VControlModeSel]</a> to select which group's status should be read back.
7:0	<b>Port0PhyStatus</b> . Read-only; updated-by-hardware. Reset: 21h. PHY Status of Port 0. Use the <a href="#">D10F0x4C_x4000_0000[VControlModeSel]</a> to select which group's status should be read back.

**D10F0x4C\_x4000\_0008 USB Common PHY Calibration and Control**

Bits	Description
31	Reserved.
30:24	<b>CommonPhyCalBus.</b> Read-only; updated-by-hardware. Reset: X. Specifies PHY common calibration bus.
23	Reserved.
22:20	<b>ComCalAmp.</b> Read-only; updated-by-hardware. Reset: X. Specifies the amplitude calibration data from PHY.
19:18	Reserved.
17	<b>AddToCommonCalibration.</b> Read-write. Reset: 1. BIOS: See 2.22.2.12. 1=The signed NewCalBus is added to the ComCalBus and returned to the PHY ports; Any overflow is clamped to all 1s; Any underflow is clamped to all 0s. 0=The signed NewCalBus replaces ComCalBus and returns to the PHY ports.
16	<b>UseCommonCalibration.</b> Read-write. Reset: 0. BIOS: See 2.22.2.12. 1=The PHY's calibration value in ComCalBus is returned to the PHY ports. 0=The value after adjustment is returned to the PHY ports.
15:8	<b>NewCalBus.</b> Read-write. Reset: 0. BIOS: See 2.22.2.12. Specifies the new calibration bus signed value.
7	Reserved.
6:0	<b>ComCalBus.</b> Read-only; updated-by-hardware. Reset: X. Specifies the common calibration bus value from PHY before adjustment.

**D10F0x4C\_x4000\_000C USB Common PHY Control**

BIOS: See 2.15.2.9 [xHC USB2.0 Common PHY Calibration].

Bits	Description										
31:24	<b>DllControl: DLL Control.</b> Read-write. Reset: A0h. Specifies USB PHY DLL control. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Bit</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>[1:0]</td> <td>Reserved.</td> </tr> <tr> <td>[2]</td> <td>DLL_EN_PFDphases. 1=Enable DLL phase-sampling based lock detection.</td> </tr> <tr> <td>[5:3]</td> <td>DLL_cpump. DLL charge pump current control.</td> </tr> <tr> <td>[7:6]</td> <td>DLL_VtoI. DLL gain control.</td> </tr> </tbody> </table>	Bit	Definition	[1:0]	Reserved.	[2]	DLL_EN_PFDphases. 1=Enable DLL phase-sampling based lock detection.	[5:3]	DLL_cpump. DLL charge pump current control.	[7:6]	DLL_VtoI. DLL gain control.
Bit	Definition										
[1:0]	Reserved.										
[2]	DLL_EN_PFDphases. 1=Enable DLL phase-sampling based lock detection.										
[5:3]	DLL_cpump. DLL charge pump current control.										
[7:6]	DLL_VtoI. DLL gain control.										
23:21	Reserved.										
20	<b>PllBypass: PLL Bypass.</b> Read-write. Reset: 0. 1=Enable USB Common PLL bypass.										
19:16	<b>DutyAdj.</b> Read-write. Reset: 4h. Specifies CLK480 duty cycle control from 40-60% to 60-40%.										
15:12	<b>PVI.</b> Read-write. Reset: 2h. Specifies PLL V-I converter control for common block PLL.										
11:8	<b>IRefAdj.</b> Read-write. Reset: 1. Specifies internal reference bias adjustment for common block.										
7:4	<b>XRefAdj.</b> Read-write. Reset: 1. Specifies external reference bias adjustment for common block.										
3:0	<b>CPAdj.</b> Read-write. Reset: 4h. Specifies charge pump setting for common block PLL.										

**D10F0x4C\_x4000\_0010 HS Loopback Test**

Bits	Description						
31:23	Reserved.						
22:21	<b>StatusOfPowerUpStateCheck: Status of ports for PowerUp State Check.</b> Read-only; updated-by-hardware. Reset: 0. This field is bit-significant. It indicates the Power Up state auto-checking result from the individual port. For each bit, 0=PHY Power Up State checking failed. 1=PHY Power Up State checking is good. <table border="1"> <thead> <tr> <th>Bit</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>[0]</td> <td>Port 0</td> </tr> <tr> <td>[1]</td> <td>Port 1</td> </tr> </tbody> </table>	Bit	Definition	[0]	Port 0	[1]	Port 1
Bit	Definition						
[0]	Port 0						
[1]	Port 1						
20	<b>EnablePHYPowerUpStateChecking.</b> Read-write. Reset: 0. 1=Enable auto-checking on PHY Power Up State. There is built-in logic to check the PHY default Power Up state to detect manufacturing defects in the PHY macro.						
19:12	<b>GoodReceivedPacketCount.</b> Read-only; updated-by-hardware. Reset: 0. Specifies the number of good packets that the host controller received during the loopback test mode. These bits will be cleared by programming EnableLoopBackTest=0.						
11	Reserved.						
10	<b>LoopbackTestDone.</b> Read-only. Reset: 0. 1=Loopback test is done.						
9	<b>LoopbackTestStatus.</b> Read-only. Reset: 0. 0=CRC error on loopback receiving data. 1=Good CRC on loopback receiving data.						
8	<b>EnableLoopBackTest.</b> Read-write. Reset: 0. 1=Enable external USB port loopback test. The loopback test is to set one port to TX mode (Test Packet mode) and one port to RX mode (Test SE0_NAK). See EHCI[3:1]x[70,6C,68,64][PortTestControl] for information on the tests.						
7:4	Reserved.						
3:0	<b>RxPktCnt: Received Packet Count.</b> Read-write. Reset: 0. This counter defines the number (in power of 16) of RX data packets that should be checked for the loop back test.						

**D10F0x4C\_x4000\_0014 CL Loopback Control**

Bits	Description								
31	<b>LoopBackTestDone.</b> Read-only; updated-by-hardware. Reset: 0. Host controller sets the bit when loop back is done. The bit is cleared when software clears the LoopBackTestStart bit. 1=Loopback test is done.								
30:16	<b>LoopBackTestStatus.</b> Read-only; updated-by-hardware. Reset: 0. This field is bit-significant. It specifies the loopback test status for the port under test. Software can only check these status bits when the LoopBackTestDone bit is set by the host controller. These bits can only be cleared when software clears the LoopBackTestStart bit. For each bit, 0=Fail. 1=Pass. <table border="1"> <thead> <tr> <th>Bit</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>[0]</td> <td>Port 0</td> </tr> <tr> <td>[1]</td> <td>Port 1</td> </tr> <tr> <td>[4:2]</td> <td>Reserved</td> </tr> </tbody> </table>	Bit	Definition	[0]	Port 0	[1]	Port 1	[4:2]	Reserved
Bit	Definition								
[0]	Port 0								
[1]	Port 1								
[4:2]	Reserved								
15:8	<b>LoopBackTestData.</b> Read-write. Reset: 0. Specifies the 1-byte test data pattern for transmit; receive logic checks the received data to match with this data pattern.								

7:4	<b>PortUnderTest</b> . Read-write. Reset: 0. Software selects the port under test through these bits. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>0001b-0000b</td> <td>Port number &lt;PortUnderTest&gt;</td> </tr> <tr> <td>1111b-0010b</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Definition	0001b-0000b	Port number <PortUnderTest>	1111b-0010b	Reserved
Bits	Definition						
0001b-0000b	Port number <PortUnderTest>						
1111b-0010b	Reserved						
3:1	Reserved.						
0	<b>LoopBackTestStart</b> . Read-write. Reset: 0. Software sets this bit to start the loopback test and clears this bit to clear out all the test status before starting the next loop. 1=Start the loopback test. 0=Clear LoopBackTestStatus and LoopBackTestDone.						

**D10F0x4C\_x4000\_0018 Misc Control**

Bits	Description
31:12	Reserved.
11:8	<b>InterPacketGapAdjustCounter</b> . Read-write. Reset: 4h. Specifies the counter used to adjust the inter-packet gap for test packet.
7:3	Reserved.
2	<b>TestPacketInterPacketGapEnable</b> . Read-write. Reset: 1. 1=Enable inter-packet gap function for general test packet mode.
1	Reserved.
0	<b>U2IFPowerSavingEnable</b> . Read-write. Reset: 1. 1=Enable power saving clock gating; dynamic clock gating is enabled when U2IF trans_state is in idle state.

**D10F0x4C\_x4000\_0020 USB SCLK Event Counter 0**

Bits	Description
31:0	<b>UsbSCLKEventCnt0Lo</b> . Read-only. Reset: 0. USB SCLK Event Counter 0[31:0].

**D10F0x4C\_x4000\_0024 USB SCLK Event Counter 1**

Bits	Description
31:0	<b>UsbSCLKEventCnt1Lo</b> . Read-only. Reset: 0. USB SCLK Event Counter 1[31:0].

**D10F0x4C\_x4000\_0028 USB SCLK Event Counter Select**

Bits	Description
31:24	<b>UsbSCLKEventCnt1Hi</b> . Read-only. Reset: 0. USB SCLK Event Counter 1[39:32].
23:16	<b>UsbSCLKEventCnt0Hi</b> . Read-only. Reset: 0. USB SCLK Event Counter 0[39:32].
15:8	<b>UsbSCLKEventSel1</b> . Read-write. Reset: 0. Selects event to be counted by USB SCLK Event Counter 1.
7:0	<b>UsbSCLKEventSel0</b> . Read-write. Reset: 0. Selects event to be counted by USB SCLK Event Counter 0.

**D10F0x4C\_x4000\_002C USB SCLK Event Counter Control**

Bits	Description
31:3	Reserved.
2	<b>UsbSCLKEventCntShadow.</b> Read-write. Reset: 0. 1=Transfer USB SCLK event counter to shadow register.
1	<b>UsbSCLKEventCntResetB.</b> Read-write. Reset: 1. 0=Resets USB SCLK event counters.
0	<b>UsbSCLKEventCntEn.</b> Read-write. Reset: 0. 1=Enable USB SCLK event counters.

**D10F0x4C\_x4000\_0048 LPM Control**

Bits	Description																				
31:18	Reserved.																				
17	<b>CfguP1VirLpmRespEn: Port1 Virtual LPM Response Enable.</b> Read-write. Reset: 0. 0=Disable virtual LPM Response. 1=Force a virtual response of port 1 LPM transaction. The type of response is determined by CfguP1VirLpmResp.																				
16:14	<b>CfguP1VirLpmResp: Port1 Virtual LPM Response.</b> Read-write. Reset: 0. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>ACK</td> <td>100b</td> <td>Reserved</td> </tr> <tr> <td>001b</td> <td>Reserved</td> <td>101b</td> <td>Reserved</td> </tr> <tr> <td>010b</td> <td>STALL</td> <td>110b</td> <td>TIMEOUT</td> </tr> <tr> <td>011b</td> <td>NYET</td> <td>111b</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Definition	Bits	Definition	000b	ACK	100b	Reserved	001b	Reserved	101b	Reserved	010b	STALL	110b	TIMEOUT	011b	NYET	111b	Reserved
Bits	Definition	Bits	Definition																		
000b	ACK	100b	Reserved																		
001b	Reserved	101b	Reserved																		
010b	STALL	110b	TIMEOUT																		
011b	NYET	111b	Reserved																		
13	<b>CfguP0VirLpmRespEn: Port0 Virtual LPM Response Enable.</b> Read-write. Reset: 0. 0=Disable virtual LPM Response. 1=Force a virtual response of port 0 LPM transaction. The type of response is determined by CfguP0VirLpmResp.																				
12:10	<b>CfguP0VirLpmResp: Port0 Virtual LPM Response.</b> Read-write. Reset: 0. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>ACK</td> <td>100b</td> <td>Reserved</td> </tr> <tr> <td>001b</td> <td>Reserved</td> <td>101b</td> <td>Reserved</td> </tr> <tr> <td>010b</td> <td>STALL</td> <td>110b</td> <td>TIMEOUT</td> </tr> <tr> <td>011b</td> <td>NYET</td> <td>111b</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Definition	Bits	Definition	000b	ACK	100b	Reserved	001b	Reserved	101b	Reserved	010b	STALL	110b	TIMEOUT	011b	NYET	111b	Reserved
Bits	Definition	Bits	Definition																		
000b	ACK	100b	Reserved																		
001b	Reserved	101b	Reserved																		
010b	STALL	110b	TIMEOUT																		
011b	NYET	111b	Reserved																		
9	<b>CfguDevRecoveryDis: Device Recovery Timer Disable.</b> Read-write. Reset: 0. 1=xHC does not wait for the device recovery time.																				

8:5	<p><b>CfguRWakeRsmDuration[3:0]: Remote Wake Resume Duration.</b> Read-write. Reset: 0. Software sets this field to indicate how long the xHC should stay in L1Resuming after remote wake is detected. The L1 Resume Timer will load this duration after remote wake is detected and HLE is 0. When the timer expires, the Root Hub state will jump to SENDEOR.</p> <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>0000b</td> <td>60 us</td> <td>1011b-0011b</td> <td>60 us + value * 75 us</td> </tr> <tr> <td>0001b</td> <td>135 us</td> <td>1100b</td> <td>960 us</td> </tr> <tr> <td>0010b</td> <td>210 us</td> <td>1111b-1101b</td> <td>Reserved</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	<u>Bits</u>	<u>Definition</u>	0000b	60 us	1011b-0011b	60 us + value * 75 us	0001b	135 us	1100b	960 us	0010b	210 us	1111b-1101b	Reserved
<u>Bits</u>	<u>Definition</u>	<u>Bits</u>	<u>Definition</u>														
0000b	60 us	1011b-0011b	60 us + value * 75 us														
0001b	135 us	1100b	960 us														
0010b	210 us	1111b-1101b	Reserved														
4:0	<p><b>CfguL1ResidencyDuration: L1 Residency Duration.</b> Read-write. Reset: 0. BIOS: 1. Software sets this field to indicate the minimum duration that the xHC should stay in L1Suspend. The L1 Residency Timer will load this duration when LPM transaction ACK is received and HLE is 0. When the timer expires and there is either Hardware Initiated L1 Resume Request or Remote Wake detected, the Root Hub state will jump to L1Resuming.</p> <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>00000b</td> <td>50 us</td> <td>11110b-00010b</td> <td>(value + 1)* 50 us</td> </tr> <tr> <td>00001b</td> <td>100 us</td> <td>11111b</td> <td>1.6 ms</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	<u>Bits</u>	<u>Definition</u>	00000b	50 us	11110b-00010b	(value + 1)* 50 us	00001b	100 us	11111b	1.6 ms				
<u>Bits</u>	<u>Definition</u>	<u>Bits</u>	<u>Definition</u>														
00000b	50 us	11110b-00010b	(value + 1)* 50 us														
00001b	100 us	11111b	1.6 ms														

**D10F0x4C\_x4000\_0050 USB Common PHY Control 2**

BIOS: See 2.15.2.9 [xHC USB2.0 Common PHY Calibration].

Bits	Description
31:4	Reserved.
3:0	<b>BgAdj.</b> Read-write. Reset: 6. Bandgap voltage adjust.

**D10F0x4C\_x4000\_0054 USB Common PHY Control 3**

BIOS: See 2.15.2.9 [xHC USB2.0 Common PHY Calibration].

Bits	Description
31:2	Reserved.
1	<b>CalEnable.</b> Read-write. Reset: 0. 1=Enable calibration.
0	Reserved.

**D10F0x50 PME Capability**

Bits	Description												
31:27	<p><b>PmeSupport.</b> Read-only. Reset: 19h. Indicates the power states in which the function may assert PME#. For each bit, 0=The function is not capable of asserting the PME# signal while in that power state.</p> <table border="1"> <thead> <tr> <th><u>Bit</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>[0]</td> <td>1=PME# can be asserted from D0</td> </tr> <tr> <td>[1]</td> <td>1=PME# can be asserted from D1</td> </tr> <tr> <td>[2]</td> <td>1=PME# can be asserted from D2</td> </tr> <tr> <td>[3]</td> <td>1=PME# can be asserted from D3hot</td> </tr> <tr> <td>[4]</td> <td>1=PME# can be asserted from D3cold</td> </tr> </tbody> </table>	<u>Bit</u>	<u>Definition</u>	[0]	1=PME# can be asserted from D0	[1]	1=PME# can be asserted from D1	[2]	1=PME# can be asserted from D2	[3]	1=PME# can be asserted from D3hot	[4]	1=PME# can be asserted from D3cold
<u>Bit</u>	<u>Definition</u>												
[0]	1=PME# can be asserted from D0												
[1]	1=PME# can be asserted from D1												
[2]	1=PME# can be asserted from D2												
[3]	1=PME# can be asserted from D3hot												
[4]	1=PME# can be asserted from D3cold												
26	<b>D2Support.</b> Read-only. Reset: 0. 0=This function doesn't support the D2 Power Management State.												

25	<b>D1Support.</b> Read-only. Reset: 0. 0=This function doesn't support the D1 Power Management State.
24:22	<b>AuxCurrent.</b> Read-only. Reset: 0. Specifies the 3.3 V auxiliary current requirements for the PCI function. Since <a href="#">D10F0x54[Data]</a> is implemented by this function, reads of this field must return a value of 000b; <a href="#">D10F0x54[Data]</a> takes precedence over this field for 3.3 V auxiliary current requirement reporting.
21	<b>DSI: Device Specific Initialization.</b> Read-only. Reset: 0. This bit indicates whether special initialization of this function is required (beyond the standard PCI configuration header) before the generic class device driver is able to use it.
20	Reserved.
19	<b>PmeClock: PME clock.</b> Read-only. Reset: 0. 0=Indicates that no PCI clock is required for the function to generate PME#.
18:16	<b>Version.</b> Read-only. Reset: 011b. 011b=Indicates that this function complies with Revision 1.2 of the PCI Power Management Interface Specification.
15:8	<b>NextItemPointer: Next Item Pointer.</b> Read-only. Reset: 70h. Points to the location of next item in the function's capability list.
7:0	<b>CapId.</b> Read-only. Reset: 1. The linked list item is for PCI Power Management.

#### **D10F0x54 PME Control / Status**

Bits	Description
31:24	<b>Data.</b> Read-only. Reset: 0. Reports the state dependent data requested by <a href="#">D10F0x54[DataSelect]</a> . The value of this field is scaled by <a href="#">D10F0x54[DataScale]</a> .
23:16	Reserved.
15	<b>PmeStatus.</b> Read; set-by-hardware; write-1-to-clear. Cold reset: 0. BIOS: See <a href="#">2.15.2.14</a> . 1=The PME# signal is asserted, which is independent of PmeEn. 0=PME# is de-asserted.
14:13	<b>DataScale.</b> Read-only. Reset: 0. Indicates the scaling factor to be used when interpreting the value of <a href="#">D10F0x54[Data]</a> . The value and meaning of this field will vary depending on which data value has been selected by <a href="#">D10F0x54[DataSelect]</a> .
12:9	<b>DataSelect.</b> Read-only. Reset: 0. Selects which data is to be reported through <a href="#">D10F0x54[Data]</a> and <a href="#">D10F0x54[DataScale]</a> .
8	<b>PmeEn.</b> Read-write. Cold reset: 0. 1=Enable the function to assert PME# if PmeStatus=1. 0=PME# assertion is disabled.
7:4	Reserved.
3	<b>NoSoftReset.</b> Read-only. Reset: 1. 1=Upon transitioning from D3hot to D0 because of PowerState, devices do not perform an internal reset ; Configuration context is preserved; No additional operating system intervention is required to preserve configuration context beyond writing the PowerState bits. 0=Upon transitioning from D3hot to D0 via software control of the PowerState bits, devices do perform an internal reset; Configuration context is lost; Full reinitialization sequence is needed to return the device to D0 Initialized. Regardless of this bit, devices that transition from D3hot to D0 by a system or bus segment reset will return to the device state D0 uninitialized with only PME context preserved if PME is supported and enabled.



2	Reserved.												
1:0	<p><b>PowerState.</b> Read-only. Reset: 0. This field determines the current power state of a function and sets the function into a new power state. If software attempts to write an unsupported, optional state to this field, the write operation must be completed normally on the bus; however, the data is discarded and no state change occurs.</p> <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>D0</td> <td>10b</td> <td>D2</td> </tr> <tr> <td>01b</td> <td>D1</td> <td>11b</td> <td>D3hot</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	<u>Bits</u>	<u>Definition</u>	00b	D0	10b	D2	01b	D1	11b	D3hot
<u>Bits</u>	<u>Definition</u>	<u>Bits</u>	<u>Definition</u>										
00b	D0	10b	D2										
01b	D1	11b	D3hot										

**D10F0x60 SBRN**

Bits	Description																				
31:24	Reserved.																				
23:20	<b>DBESLD: Default Best Effort Service Latency Deep.</b> Read-only. Reset: 0h. Vendor Defined.																				
19:16	<b>DBESL: Default Best Effort Service Latency.</b> Read-only. Reset: 0h. Vendor Defined.																				
15:14	Reserved.																				
13:8	<p><b>FLAdj: Frame Length Timing Value.</b> Read-only. Reset: 20h. Each decimal value change to this register corresponds to 16 high-speed bit times. The SOF cycle time (number of SOF counter clock periods to generate a SOF microframe length)=59488 + 16*FLAdj.</p> <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Frame Length (# HS bit times)</u></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>59488</td> </tr> <tr> <td>1</td> <td>59504</td> </tr> <tr> <td>2</td> <td>59520</td> </tr> <tr> <td>...</td> <td>...</td> </tr> <tr> <td>31</td> <td>59984</td> </tr> <tr> <td>32</td> <td>60000</td> </tr> <tr> <td>...</td> <td>...</td> </tr> <tr> <td>62</td> <td>60480</td> </tr> <tr> <td>63</td> <td>60496</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Frame Length (# HS bit times)</u>	0	59488	1	59504	2	59520	...	...	31	59984	32	60000	...	...	62	60480	63	60496
<u>Bits</u>	<u>Frame Length (# HS bit times)</u>																				
0	59488																				
1	59504																				
2	59520																				
...	...																				
31	59984																				
32	60000																				
...	...																				
62	60480																				
63	60496																				
7:0	<b>SBRN: Serial Bus Specification Release Number.</b> Value: 30h. Specifies the release of the Universal Serial Bus Specification with which this Universal Serial Bus Host Controller module is compliant with.																				

**D10F0x70 MSI Control**

Bits	Description
31:25	Reserved.
24	<b>PVMC: Per-vector masking capable.</b> Read-only. Reset: 0. 1=The function supports MSI per-vector masking. 0=The function does not support MSI per-vector masking.
23	<b>C64: 64-bit Address Capable.</b> Read-only. Reset: 1. 1=The function is capable of sending a 64-bit message address. 0=The function is not capable of sending a 64-bit message address.

22:20	<b>MME: Multiple Message Enable.</b> Read-write. Reset: 0. System software writes to this field to indicate the number of allocated messages. <table border="1"> <thead> <tr> <th>Bits</th> <th># of messages allocated</th> <th>Bits</th> <th># of messages allocated</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>1</td> <td>100b</td> <td>16</td> </tr> <tr> <td>001b</td> <td>2</td> <td>101b</td> <td>32</td> </tr> <tr> <td>010b</td> <td>4</td> <td>111b-110b</td> <td>Reserved</td> </tr> <tr> <td>011b</td> <td>8</td> <td></td> <td></td> </tr> </tbody> </table>	Bits	# of messages allocated	Bits	# of messages allocated	000b	1	100b	16	001b	2	101b	32	010b	4	111b-110b	Reserved	011b	8		
Bits	# of messages allocated	Bits	# of messages allocated																		
000b	1	100b	16																		
001b	2	101b	32																		
010b	4	111b-110b	Reserved																		
011b	8																				
19:17	<b>MMC: Multiple Message Capable.</b> Read-only. Reset: 011b. System software reads this field to determine the number of requested messages. 011b=8 messages requested.																				
16	<b>MSIE: MSI Enable.</b> Read-write. Reset: 0. 1=MSI is enabled and INTx# pins are not used to generate interrupts. 0=MSI is disabled and the traditional pins are used to request service.																				
15:8	<b>NextItemPointer: Next Item Pointer.</b> Read-only. Reset: 90h. Points to next the capability structure.																				
7:0	<b>Capability.</b> Read-only. Reset: 5h. MSI USB ID.																				

**D10F0x74 MSI Address**

Bits	Description
31:2	<b>MsiAddress: MSI Address.</b> Read-write. Reset: 0. Specifies the lower bits of system-specified message address [31:2]. Always doubleword aligned.
1:0	Reserved.

**D10F0x78 MSI Upper Address**

Bits	Description
31:0	<b>MsiUpperAddress: MSI Upper Address.</b> Read-write. Reset: 0. Specifies the system-specified message upper address [61:32].

**D10F0x7C MSI Data**

Bits	Description
15:0	<b>MsiData: MSI Data.</b> Read-write. Reset: 0. System-specified message.

**D10F0x80 MSI Mask Bits**

Bits	Description
31:0	<b>MaskBits: Mask Bits.</b> Read-write. Reset: 0. For each mask bit that is set, the function is prohibited from sending the associated message.

**D10F0x90 MSI-X Control**

Bits	Description
31	<b>MsiXEnable: MSI-X Enable.</b> Read-write. Reset: 0.

30	<b>FunctionMask: Function Mask.</b> Read-write. Reset: 0.
29:27	Reserved.
26:16	<b>TableSize: Table Size.</b> Read-only. Reset: 7h.
15:8	<b>NextItemPointer: Next Item Pointer.</b> Read-only. Reset: A0h.
7:0	<b>CapabilityID: Capability ID.</b> Read-only. Reset: 11h.

#### D10F0x94 MSI-X Table Offset/Table BIR

Bits	Description																				
31:3	<b>TableOffset: Table Offset.</b> Read-only. Reset: 200h. Used as an offset from the address contained by one of the function's Base Address registers to point to the base of the MSI-X Table. The lower 3 TableBIR bits are masked off (set to zero) by software to form a 32-bit QWORD-aligned offset.																				
2:0	<b>TableBIR: Table BIR.</b> Read-only. Reset: 0. Indicates which Base Address register, starting at 10h in the configuration space, is used to map the function's MSI-X table into memory space. For a 64-bit Base Address register, the TableBIR indicates the lower DWORD. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Bits</th> <th>Base Address Register</th> <th>Bits</th> <th>Base Address Register</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>10h</td> <td>4h</td> <td>20h</td> </tr> <tr> <td>1h</td> <td>14h</td> <td>5h</td> <td>24h</td> </tr> <tr> <td>2h</td> <td>18h</td> <td>7h-6h</td> <td>Reserved</td> </tr> <tr> <td>3h</td> <td>1Ch</td> <td></td> <td></td> </tr> </tbody> </table>	Bits	Base Address Register	Bits	Base Address Register	0h	10h	4h	20h	1h	14h	5h	24h	2h	18h	7h-6h	Reserved	3h	1Ch		
Bits	Base Address Register	Bits	Base Address Register																		
0h	10h	4h	20h																		
1h	14h	5h	24h																		
2h	18h	7h-6h	Reserved																		
3h	1Ch																				

#### D10F0x98 MSI-X PBA Offset/PBA BIR

Bits	Description
31:3	<b>PBAOffset: PBA Offset.</b> Read-only. Reset: 210h. Used as an offset from the address contained by one of the function's Base Address registers to point to the base of the MSI-X PBA. The lower 3 PBA BIR bits are masked off (set to zero) by software to form a 32-bit QWORD-aligned offset.
2:0	<b>PBABIR: PBA BIR.</b> Read-only. Reset: 0. Indicates which Base Address register, starting at 10h in the configuration space, is used to map the functions MSI-X PBA into memory space. See: <a href="#">D10F0x94[TableBIR]</a> .

#### D10F0xA0 PCIe® Capability List

Bits	Description
31:30	Reserved.
29:25	<b>IntMsgNum: Interrupt Message Number.</b> Read-only. Reset: 0. This field indicates which MSI/MSI-X vector is used for the interrupt message generated in association with any of the status bits of this Capability structure.
24	<b>SlotImplemented: Slot Implemented.</b> Value: 0.
23:20	<b>DevicePortType: Device/Port Type.</b> Value: 9h. 9h=Indicates Root Complex Integrated Endpoint.
19:16	<b>CapabilityVersion: Capability Version.</b> Value: 2h. Indicates PCI-SIG defined PCI Express® Capability structure version number. 2h=PCI Express® Base Rev 2.0.

15:8	<b>NextItemPointer: Next Item Pointer.</b> Value: 0. 00h=End of linked list capabilities.
7:0	<b>CapabilityID: Capability ID.</b> Read-only. Reset: 10h. PCI Express® Capability.

**D10F0xA4 Device Capability**

Bits	Description										
31:29	Reserved.										
28	<b>FLRCapability: Function Level Reset Capability.</b> Read-only. Reset: 0. 0=Function Level Reset is not supported.										
27:26	<b>SlotPowerLimitScale: Captured Slot Power Limit Scale (Upstream Ports only).</b> Value: 0. Specifies the scale used for the SlotPowerLimitValue. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>1.0 x</td> </tr> <tr> <td>01b</td> <td>0.1 x</td> </tr> <tr> <td>10b</td> <td>0.01 x</td> </tr> <tr> <td>11b</td> <td>0.001 x</td> </tr> </tbody> </table>	Bits	Definition	00b	1.0 x	01b	0.1 x	10b	0.01 x	11b	0.001 x
Bits	Definition										
00b	1.0 x										
01b	0.1 x										
10b	0.01 x										
11b	0.001 x										
25:18	<b>SlotPowerLimitValue: Captured Slot Power Limit Value (Upstream Ports only).</b> Read-only. Reset: 0. In combination with SlotPowerLimitScale, specifies the upper limit on power supplied by slot. Power limit (in Watts)=SlotPowerLimitValue * SlotPowerLimitScale.										
17:16	Reserved.										
15	<b>RoleBasedErrorReport: Role-Based Error Reporting.</b> Value: 1. 1=Indicates that the function implements the functionality originally defined in the Error Reporting ECN for PCI Express® Base Specification, Revision 1.0a, and later incorporated into PCI Express® Base Specification, Revision 1.1.										
14:12	Reserved.										
11:9	<b>EndpointL1AcceptableLatency: Endpoint L1 Acceptable Latency.</b> Value: 111b. This field indicates the acceptable latency that an Endpoint can withstand due to the transition from L1 state to the L0 state. It is essentially an indirect measure of the Endpoint's internal buffering. Power management software uses the reported L1 Acceptable Latency number to compare against the L1 Exit Latencies reported (see below) by all components comprising the data path from this Endpoint to the Root Complex Root Port to determine whether ASPM L1 entry can be used with no loss of performance. 111b=No limit.										
8:6	<b>EndpointL0sAcceptableLatency: Endpoint L0s Acceptable Latency.</b> Value: 111b. This field indicates the acceptable total latency that an Endpoint can withstand due to the transition from L0s state to the L0 state. It is essentially an indirect measure of the Endpoints internal buffering. Power management software uses the reported L0s Acceptable Latency number to compare against the L0s exit latencies reported by all components comprising the data path from this Endpoint to the Root Complex Root Port to determine whether ASPM L0s entry can be used with no loss of performance. 111b=No limit.										
5	<b>ExtTagFieldSup: Extended Tag Field Supported.</b> Value: 0. This bit indicates the maximum supported size of the Tag field as a Requester. 0=Tag field size of 5-bits supported.										

4:3	<b>PhantomFuncSup: Phantom Functions Supported.</b> Value: 0. 0=No Function Number bits are used for Phantom Functions.			
2:0	<b>MaxPayloadSizeSup: Max Payload Size Supported.</b> Value: 0. This field indicates the maximum payload size that the function can support for TLPs.			
	<u>Bits</u>	<u>Payload Size</u>	<u>Bits</u>	<u>Payload Size</u>
	0h	128 bytes max payload size	4h	2048 bytes max payload size
	1h	256 bytes max payload size	5h	4096 bytes max payload size
	2h	512 bytes max payload size	7h-6h	Reserved
	3h	1024 bytes max payload size		

**D10F0xA8 Device Control/Status**

Bits	Description
31:22	Reserved.
21	<b>TransPending: Transactions Pending.</b> Read-only. Reset: 0. 1=The function has issued Non-Posted Requests which have not been completed. 0=All outstanding Non-Posted Requests have completed or have been terminated by the Completion Timeout mechanism. This bit must also be cleared upon the completion of an FLR.
20	<b>AUXPowerDetected: AUX Power Detected.</b> Read-only. Reset: 0. 1=AUX power is detected by the function.
19	<b>UnsupReqDetected: Unsupported Request Detected.</b> Read; set-by-hardware; write-1-to-clear. Reset: 0. 1=The function received an Unsupported Request. Errors are logged in this register regardless of UnsupReqRptEn.
18	<b>FatalErrDetected: Fatal Error Detected.</b> Read; set-by-hardware; write-1-to-clear. Reset: 0. 1=Fatal errors are detected. Errors are logged in this register regardless of FatalErRptEn.
17	<b>NonFatalErrDetected: Non-Fatal Error Detected.</b> Read; set-by-hardware; write-1-to-clear. Reset: 0. 1=Nonfatal errors are detected. Errors are logged in this register regardless of NonFatalErrRptEn.
16	<b>CorrErrDetected: Correctable Error Detected.</b> Read; set-by-hardware; write-1-to-clear. Reset: 0. 1=Correctable errors are detected. Errors are logged in this register regardless of CorrErrRptEn.
15	<b>InitiateFLR: Initiate Function Level Reset.</b> RAZ; write-1-only. Reset: 0. FLR is not supported.
14:12	<b>MaxReadRequestSize.</b> Read-write. Reset: 010b. This field sets the maximum Read Request size for the function as a Requester. The function must not generate Read Requests with size exceeding the set value. 010b=512 bytes maximum Read Request size.
11	<b>EnNoSnoop: Enable No Snoop.</b> Read-write. Reset: 1. 1=Function is permitted to set the No Snoop bit in the Requester Attributes of transactions that it initiates and do not require hardware enforced cache coherency.
10	<b>AuxPowerPMEn: Auxiliary Power PM Enable.</b> Read-write. Cold reset: 0. 1=Enable a function to draw auxiliary (AUX) power independent of PME AUX power. Functions that require AUX power on legacy operating systems should continue to indicate PME AUX power requirements. AUX power is allocated as requested in D10F0x50[AuxCurrent] independent of D10F0x54[PmeEn]. Functions that consume AUX power must preserve the value of this sticky register when AUX power is available and preserve the value through Conventional Reset.
9	<b>PhantomFunctionsEn: Phantom Functions Enable.</b> Value: 0. 0=The Function is not allowed to use Phantom Functions.
8	<b>ExtTagFieldEn: Extended Tag Field Enable.</b> Value: 0. 0=Function is restricted to a 5-bit Tag field.

7:5	<b>MaxPayloadSize.</b> Read-write. Reset: 0. This field sets maximum TLP payload size for the function. as a receiver, the function must handle TLPs as large as the set value. As a Transmitter, the Function must not generate TLPs exceeding the set value. Permissible values that can be programmed are indicated by <b>D10F0xA4[MaxPayloadSizeSup]</b> . <table border="0"> <thead> <tr> <th><u>Bits</u></th> <th><u>Payload Size</u></th> <th><u>Bits</u></th> <th><u>Payload Size</u></th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>128 bytes max payload size</td> <td>4h</td> <td>2048 bytes max payload size</td> </tr> <tr> <td>1h</td> <td>256 bytes max payload size</td> <td>5h</td> <td>4096 bytes max payload size</td> </tr> <tr> <td>2h</td> <td>512 bytes max payload size</td> <td>7h-6h</td> <td>Reserved</td> </tr> <tr> <td>3h</td> <td>1024 bytes max payload size</td> <td></td> <td></td> </tr> </tbody> </table>	<u>Bits</u>	<u>Payload Size</u>	<u>Bits</u>	<u>Payload Size</u>	0h	128 bytes max payload size	4h	2048 bytes max payload size	1h	256 bytes max payload size	5h	4096 bytes max payload size	2h	512 bytes max payload size	7h-6h	Reserved	3h	1024 bytes max payload size		
<u>Bits</u>	<u>Payload Size</u>	<u>Bits</u>	<u>Payload Size</u>																		
0h	128 bytes max payload size	4h	2048 bytes max payload size																		
1h	256 bytes max payload size	5h	4096 bytes max payload size																		
2h	512 bytes max payload size	7h-6h	Reserved																		
3h	1024 bytes max payload size																				
4	<b>EnRelaxOrder: Enable Relaxed Ordering.</b> Read-write. Reset: 1. 1= Function is permitted to set the Relaxed Ordering bit in the Attributes field of transactions that it initiates and do not require strong write ordering.																				
3	<b>UnsupReqRptEn: Unsupported Request Reporting Enable.</b> Read-write. Reset: 0. This bit, in conjunction with other bits, controls the signaling of Unsupported Requests by sending Error Messages.																				
2	<b>FatalErrRptEn: Fatal Error Reporting Enable.</b> Read-write. Reset: 0. This bit, in conjunction with other bits, controls sending ERR_FATAL Messages																				
1	<b>NonFatalErrRptEn: Non-Fatal Error Reporting Enable.</b> Read-write. Reset: 0. This bit, in conjunction with other bits, controls sending ERR_NONFATAL Messages																				
0	<b>CorrErrRptEn: Correctable Error Reporting Enable.</b> Read-write. Reset: 0. This bit, in conjunction with other bits, controls sending ERR_COR Messages																				

### D10F0xC4 Device Capabilities 2

Bits	Description
31:12	Reserved.
11	<b>LTRMechanismSupported: LTR Mechanism Supported.</b> Read-only. Reset: 1. 1=Latency Tolerance Reporting (LTR) mechanism capability is supported.
10:5	Reserved.
4	<b>CompletionTimeoutDisableSupported: Completion Timeout Disable Supported.</b> Read-only. Reset: 1. 1=The Completion Timeout Disable mechanism is supported.
3:0	<b>CompletionTimeoutRangesSupported: Completion Timeout Ranges Supported.</b> Read-only. Reset: 0. 0=Completion Timeout programmability mechanism is not supported.

### D10F0xC8 Device Control/Status 2

Bits	Description
31:11	Reserved.
10	<b>LTRMechanismEnable: LTR Mechanism Enable.</b> Read-write. Reset: 0. 1=Enable the Latency Tolerance Reporting (LTR) mechanism.
9:5	Reserved.
4	<b>CompletionTimeoutDisable: Completion Timeout Disable.</b> Read-write. Reset: 0. 1=Disable the Completion Timeout mechanism.
3:0	<b>CompletionTimeoutValue: Completion Timeout Value.</b> Read-only. Reset: 0.

**D10F0x100 LTR Extended Capability Header**

Bits	Description
31:20	<b>NextCapabilityOffset: Next Capability Offset.</b> Read-only. Reset: 00h. 00h=End of capability list.
19:16	<b>CapabilityVersion: Capability Version.</b> Read-only. Reset: 1.
15:0	<b>PCIExpressExtendedCapabilityID: PCI Express® Extended Capability ID.</b> Read-only. Reset: 18h. Extended Capability ID for Latency Tolerance.

**D10F0x104 Max Latency**

Bits	Description
31:29	Reserved.
28:26	<b>MaxNoSnoopLatencyScale: Max No-Snoop Latency Scale.</b> Read-only. Reset: 0. This register provides a scale for the value contained within MaxNoSnoopLatencyValue. Encoding is the same as the Latency Scale fields in the LTR Message.
25:16	<b>MaxNoSnoopLatencyValue: Max No-Snoop Latency Value.</b> Read-only. Reset: 0. Along with MaxNoSnoopLatencyScale, this register specifies the maximum no-snoop latency that a device is permitted to request. Software should set this to the platform's maximum supported latency or less.
15:13	Reserved.
12:10	<b>MaxSnoopLatencyScale: Max Snoop Latency Scale.</b> Read-only. Reset: 0. This register provides a scale for the value contained within MaxSnoop LatencyValue. Encoding is the same as the Latency Scale fields in the LTR Message.
9:0	<b>MaxSnoopLatencyValue: Max Snoop Latency Value.</b> Read-only. Reset: 0. Along with MaxSnoopLatencyScale, this register specifies the maximum no-snoop latency that a device is permitted to request. Software should set this to the platform's maximum supported latency or less.

**3.25.4.3.1.3 USB xHCI Capability Registers**

The base address of the xHCI Capability register space is specified by {D10F0x14[Bar1], D10F0x10[Bar0], 0h}.

**XHCI\_CAPx02 HC Interface Version**

Bits	Description
15:0	<b>HCIVersion: Host Controller Interface Version Number.</b> Read-only. Reset: 96h. Contains a BCD encoding of the xHCI specification revision number supported by this host controller. The most significant byte of this register represents a major revision and the least significant byte is the minor revision. e.g. 0100h corresponds to xHCI version 1.0. Writing to XHCI_PMx30[HCIVersion10En] updates this field.

### 3.25.4.3.2 xHCI Power Management Registers

xHCI power management registers are accessed through the AcpiMmio region, which can be memory mapped or IO mapped. They range from `PMx24[AcpiMmioAddr]+1C00h` to `PMx24[AcpiMmioAddr]+1CFFh`. See [PMx24 \[AcpiMmioEn\]](#). These registers reside in USB 3.0 and can hold their values during S3 or S5 state if the xHC host is programmed into D3 state before entering into S3/S5 state.

#### XHCI\_PMx00 xHCI Config 0

Bits	Description
31	Reserved.
30	<b>FwPreloadComplete: Firmware Preload Complete.</b> Read-only; updated-by-hardware. Reset: 0. 1=Firmware preload is complete. 0=FwPreloadStart is cleared by software.
29	<b>FwPreloadStart: Firmware Preload Start.</b> Read-write. Reset: 0. BIOS: See 2.15.2.13. When set from 0 to 1, firmware preload will be initiated. Upon completion of firmware preload (FwPreloadComplete==1), this field should be programmed to 0.
28	<b>FwLoadMode: Firmware Load Mode.</b> Read-write. Reset: 1. Controls whether the firmware will execute a bootstrap routine to load firmware into instruction ram, or whether instruction ram is to be preloaded. 0=xHCI will execute bootstrap routine. 1=xHCI will not execute bootstrap routine.
27	Reserved.
26	<b>Xhc0BlmCgDis: xHC0 BLM Clock Gating Disable.</b> Read-write. Reset: 0. BIOS: 0. IF (U3CoreReset==0) THEN clock gating takes effect. ENDIF. 1=Disable xHC0 BLM Clock Gating.
25	<b>AlGcgDis: A-Link Global Clock Gating Disable.</b> Read-write. Reset: 0. BIOS: 0. 1=Disable USB3.0 A-Link Global Clock Gating.
24	<b>BlGcgDis: B-Link Global Clock Gating Disable.</b> Read-write. Reset: 0. BIOS: 0. 1=Disable USB3.0 B-Link Global Clock Gating.
23:22	Reserved.
21	<b>XhcSmibEn: xHC SMIB Enable.</b> Read-write. Reset: 0. BIOS: 1. 1=Enable XHC_SMIB to ACPI controller.
20:17	Reserved.
16:13	Reserved.
12	Reserved.
11	<b>Xhc0Reset: xHC0 Reset.</b> Read-write. Reset: 0. 1=Reset xHC0 core logic.
10	<b>U3CoreReset: USB3 Core Reset.</b> Read-write. Reset: 1. BIOS: See 2.15.2.13. 1=Reset the USB3.0/xHCI (xHC0) core logic. 0=xHC clock gating enable.
9	<b>U3pPhyReset: USB3 PHY Reset.</b> Read-write. Reset: 1. 1=Reset SSPHYIF power control logic in 125 MHz clock domain.
8	<b>U3pPllReset: USB3 PHY PLL Reset.</b> Read-write. Reset: 1. 1=Reset USB3.0 PHY PLL and SSPHYIF power control logic in LFPS clock domain.
7	<b>U3pLock: USB3 PHY PLL Lock.</b> Read-only. Reset: 0. 1=USB3.0 PHY PLL is locked.
6:2	Reserved.
1	Reserved.
0	<b>Xhci0Enable.</b> Read-write. Reset: 0. BIOS: See 2.15.2.4. 1=Enable xHCI0.



**XHCI\_PMx04 xHCI Firmware Addr 0**

Bits	Description
31	<b>XhciFwPreloadType.</b> Read-write. Reset: 0. BIOS: See 2.15.2.13. Controls the destination of the pre-load operation. 0=Instruction RAM. 1=Boot RAM.
30:17	Reserved.
16:0	<b>XhciFwRomAddr.</b> Read-write. Reset: 0. BIOS: See 2.15.2.13. Specifies the address of the first byte of application firmware in external ROM.

**XHCI\_PMx08 xHCI Firmware Addr 1**

Bits	Description
31:16	<b>XhciFwSize: xHCI Firmware Size.</b> Read-write. Reset: 0. BIOS: See 2.15.2.13. Specifies the size of the application firmware in the external ROM.
15:0	<b>XhciFwRamAddr: xHCI Firmware Start Address.</b> Read-write. Reset: 0. BIOS: See 2.15.2.13. Specifies the address where the first byte of application firmware is to be stored in boot RAM or instruction RAM.

**XHCI\_PMx10 xHCI Memory Config**

Bits	Description																
31:23	Reserved.																
22:16	Reserved.																
15:8	<b>CcuMode: Clock Control Mode.</b> Read-write. Reset: 07h. BIOS: 3Bh. This field contains a set of bits which, when set, enable individual features of the clock control unit. <table border="1"> <thead> <tr> <th>Bit</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>[0]</td> <td>1=Enable power control mode.</td> </tr> <tr> <td>[1]</td> <td>1=Enable SuperSpeed remote wake mode.</td> </tr> <tr> <td>[2]</td> <td>1=Enable PCLK control mode.</td> </tr> <tr> <td>[3]</td> <td>1=Enable Memory Sleep Sequencing.</td> </tr> <tr> <td>[4]</td> <td>1=Enable Stop CPU mode.</td> </tr> <tr> <td>[5]</td> <td>1=Enable firmware clock control to start firmware after PCI reset is released</td> </tr> <tr> <td>[7:6]</td> <td>Reserved.</td> </tr> </tbody> </table>	Bit	Definition	[0]	1=Enable power control mode.	[1]	1=Enable SuperSpeed remote wake mode.	[2]	1=Enable PCLK control mode.	[3]	1=Enable Memory Sleep Sequencing.	[4]	1=Enable Stop CPU mode.	[5]	1=Enable firmware clock control to start firmware after PCI reset is released	[7:6]	Reserved.
Bit	Definition																
[0]	1=Enable power control mode.																
[1]	1=Enable SuperSpeed remote wake mode.																
[2]	1=Enable PCLK control mode.																
[3]	1=Enable Memory Sleep Sequencing.																
[4]	1=Enable Stop CPU mode.																
[5]	1=Enable firmware clock control to start firmware after PCI reset is released																
[7:6]	Reserved.																
7:3	Reserved.																
2	Reserved.																
1	<b>XhcNCpuMemSlpDis: Non-CPU Memory Sleep Disable.</b> Read-write. Reset: 0. Controls whether the sleep feature is disabled on xHCI Non-Cpu internal SRAMs. 0=Do not disable sleep. 1=Disable sleep.																
0	<b>XhcCpuMemSlpDis: CPU Memory Sleep Disable.</b> Read-write. Reset: 0. Controls whether the sleep feature is disabled on xHCI Embedded CPU internal SRAMs. 0=Do not disable sleep. 1=Disable sleep.																

**XHCI\_PMx14 PLL Control**

Bits	Description
31:2	Reserved.
1	<b>U2ShutdnPIIEn: U2 Shutdown SSPHY PLL Enable.</b> Read-write. Reset: 0. Controls shutdown SSPHY PLL Enable when LTSSM is in U2 state. 1=Enable. 0=Disable.
0	<b>P3ShutdnPIIEn: P3 Shutdown SSPHY PLL Enable.</b> Read-write. Reset: 0. Controls shutdown SSPHY PLL Enable when SSPHY is in P3 state. 1=Enable. 0=Disable.

**XHCI\_PMx20 USB2.0 Wake Control**

Bits	Description
31:26	Reserved.
25:24	<b>SysBW: System Bandwidth.</b> Read-write. Reset: 0. BIOS: 1. This field is used to provide information to the controller about the available system bandwidth.
23:22	<b>U2SE1DisconMode: xHC USB2 Interface SE1 Disconnect Mode.</b> Read-write. Reset: 1. 0=Take SE1 > 2.5us as a disconnect. 1=Do not take SE1 as a disconnect.
21	Reserved.
20:18	Reserved.
17:16	Reserved.
15	Reserved.
14:13	Reserved.
12:0	Reserved.

**XHCI\_PMx30 xHCI 1.0 Enable**

Bits	Description
31:16	Reserved.
15	<b>HCIVersion10En.</b> Read-write. Reset: 0. BIOS: 1. Controls the host controller interface version number reported in <a href="#">XHCI_CAPx02</a> . 0=HCI Version is 0096h. 1=HCI Version is 0100h.
14	<b>MSEFrameIdEn: Missed Server Error Frame ID Enable.</b> Read-write. Reset: 0. BIOS: 1. 1=Enable a clarification in xHCI 1.0 specifying the conditions under which a Missed Service Error should be detected based on Frame ID. 0=Disable.
13:12	Reserved.
11	<b>SoftRetryEn: Soft Retry Enable.</b> Read-write. Reset: 0. BIOS: 1. 0=Do not support soft retry. 1=Support soft retry.
10	<b>SkipMsIocEvtEn: Skip Missed Service IOC Event Enable.</b> Read-write. Reset: 0. BIOS: 1. 1=Enable the parsing of TRBs during re-sync of an ISO pipe following detection of a Missed Service Error. 0=Disable. Always enable this bit when SkipTrbIocEvtEn==1.

9	<b>EpStateUpdateChangeEn: EPState Update Change Enable.</b> Read-write. Reset: 0. BIOS: 1. 1=Enable a clarification in xHCI 1.0 specifying that when an endpoint transitions from Stopped to Running due to a doorbell ring, the EP State shall be updated to Running before any Transfer Events are generated. 0=Disable.
8	<b>CASEn: Cold Attach Status Enable.</b> Read-write. Reset: 0. BIOS: 1. 1=Enable the xHCI 1.0 feature Cold Attach Status flag. 0=Disable.
7	<b>SbdCapObsolete: SBD Reporting Capability Obsolete.</b> Read-write. Reset: 0. BIOS: 1. 1=Enable xHCI 1.0 feature which makes Secondary Bandwidth Domain Reporting mandatory. 0=Disable.
6	<b>SkipTrbIocEvtEn: Skip TRB IOC Event Enable.</b> Read-write. Reset: 0. BIOS: 1. 1=Enable xHCI 1.0 features related to the parsing of TRBs and generation of events while skipping over TRBs due to errors and short packets. 0=Disable.
5	Reserved.
4	Reserved.
3	<b>SwLpmEn: Software LPM Enable.</b> Read-write. Reset: 0. BIOS: 1. 0=Disable software controlled LPM feature. 1=Enable software controlled LPM feature.
2	<b>FseEn: FSE Enable.</b> Read-write. Reset: 0. BIOS: 1. 0=Disable FSE feature. 1=Enable FSE feature.
1	<b>IntBlockEn: Interrupt Blocking Enable.</b> Read-write. Reset: 0. BIOS: 1. 0=Disable interrupt blocking. 1=Enable interrupt blocking.
0	<b>Xhci10En: xHCI 10 Enable.</b> Read-write. Reset: 0. Global enable bit for xHCI 1.0 features. All xHCI 1.0 features may be enabled by setting this bit, or can be individually enabled by setting the individual feature enables. 1=Enable all xHCI 1.0 features.

### **XHCI\_PMx48 SSPHY ACPI Indirect Index**

Bits	Description
31:8	Reserved.
7:2	<b>IndirectIndex: Indirect Index.</b> Read-write. Reset: 00h.
1:0	Reserved.

### **XHCI\_PMx4C SSPHY ACPI IndirectData**

Bits	Description
31:0	<b>IndirectData: Indirect Data.</b> Read-write. Reset: 0000_0000h.

### **XHCI\_PMx8C SSPHY Common Control 0**

Bits	Description
31:25	Reserved.
24	<b>CrPllCalibEn.</b> Read-write. Reset: 0h. BIOS: See 2.15.2.10. 1=Auto calibration is enabled; use calibrated value as the VCO setting. 0=Auto calibration is disabled; Use CP_PLL_VCO_TUNE[3:0] as the VCO setting if auto calibration has never been enabled; Use previously calibrated value as the VCO setting if auto calibration has been enabled once.
23:20	Reserved.

19:16	<b>PLLVcoTune[3:0]</b> . Read-write. Reset: 0h. BIOS: See 2.15.2.10. Specifies PLL VCO range initial setting (in Thermometer code). Valid settings are 0000b, 0001b, 0011b, 0111b, and 1111b. Programming invalid values may result in unpredictable behavior.
15:0	Reserved.

**XHCI\_PMxA0 SPI BAR0**

Bits	Description
31:16	<b>XhcSpiDataType0Size</b> . Read-write. Reset: 0. Type 0 data size.
15:0	<b>XhcSpiDataType0Addr</b> . Read-write. Reset: 0. Type0 data ROM address.

**XHCI\_PMxA4 SPI BAR1**

Bits	Description
31:16	<b>XhcSpiDataType1Size</b> . Read-write. Reset: 0. Type 1 data size.
15:0	<b>XhcSpiDataType1Addr</b> . Read-write. Reset: 0. Type 1 data ROM address.

**XHCI\_PMxA8 SPI BAR2**

Bits	Description
31:16	<b>XhcSpiDataType2Size</b> . Read-write. Reset: 0. Type 2 data size.
15:0	<b>XhcSpiDataType2Addr</b> . Read-write. Reset: 0. Type 2 data ROM address.

**XHCI\_PMxAC SPI BAR3**

Bits	Description
31:16	<b>XhcSpiDataType3Size</b> . Read-write. Reset: 0. Type 3 data size.
15:0	<b>XhcSpiDataType3Addr</b> . Read-write. Reset: 0. Type 3 data ROM address.

**XHCI\_PMxB0 SPI Valid Base**

Bits	Description
31:26	<b>SpiBase3</b> . Read-write. Reset: 0. Specifies the offset in SPI data block of type 3 data.
25	Reserved.
24	<b>SpiBar3Vld: SPI BAR3 Valid</b> . Read-write. Reset: 0. 1=SPI_BAR4 register is valid.
23:18	<b>SpiBase2</b> . Read-write. Reset: 0. Specifies the offset in SPI data block of type 2 data.
17	Reserved.
16	<b>SpiBar2Vld: SPI BAR2 Valid</b> . Read-write. Reset: 0. 1=SPI_BAR2 register is valid.
15:10	<b>SpiBase1</b> . Read-write. Reset: 0. Specifies the offset in SPI data block of type 1 data.
9	Reserved.

8	<b>SpiBar1Vld: SPI BAR1 Valid.</b> Read-write. Reset: 0. 1=SPI_BAR1 register is valid.
7:2	<b>SpiBase0.</b> Read-write. Reset: 0. Specifies the offset in SPI data block of type 0 data.
1	Reserved.
0	<b>SpiBar0Vld: SPI BAR0 Valid.</b> Read-write. Reset: 0. 1=SPI_BAR0 register is valid.

#### **XHCI\_PMxB4 SPI Misc**

---

Bits	Description
31:16	Reserved.
15:0	<b>XhcSpiFwId: SPI Firmware ID.</b> Read-write. Reset: 0. Firmware Version ID.

#### **XHCI\_PMx[FC:C0] SPI Data Block N**

---

Bits	Description
31:0	<b>XhcSpiDataBlock.</b> Read-write. Reset: 0. SPI Data Block.

### 3.25.5 HD Audio Controller

#### 3.25.5.1 Device 14h Function 2 (Audio Controller) Configuration Registers

##### D14F2x00 Device/Vendor ID

Bits	Description
31:16	<b>DeviceID: device ID.</b> Read-only. Reset: 780Dh.
15:0	<b>VendorID: vendor ID.</b> Read-only. Reset: 1022h.

##### D14F2x04 Status/Command

Reset: 0010\_0000h.

Bits	Description
31:30	Reserved.
29	<b>ReceivedMasterAbort: received master abort.</b> Read; write-1-to-clear. 1=A completion with an unsupported request completion status was received.
28:21	Reserved.
20	<b>CapList: capability list.</b> Read-only. 1=capability list supported.
19	<b>IntStatus: interrupt status.</b> Read-only; updated-by-hardware. 1=INTx interrupt message pending. This bit is not set by MSI
18:11	Reserved.
10	<b>IntDis: interrupt disable.</b> Read-write. 1=INTx interrupt messages generation disabled. This bit does not affect the generation of MSI.
9:3	Reserved.
2	<b>BusMasterEn: bus master enable.</b> Read-write. 1=Memory read and write request generation enabled.
1	<b>MemAccessEn: IO access enable.</b> Read-write. This bit controls if memory accesses targeting this device are accepted. 1=Enabled. 0=Disabled.
0	Reserved.

##### D14F2x08 Class Code/Revision ID

Bits	Description
31:24	<b>BaseClassCode.</b> Read-only. Reset: 04h.
23:16	<b>SubClassCode.</b> Read-only. Reset: 03h.
15:8	<b>ProgramInterface.</b> Read-only. Reset: 00h.
7:0	<b>RevID: revision ID.</b> Read-only. Reset: 03h.

**D14F2x0C Header Type**

Bits	Description
31:24	<b>BIST</b> . Value: 0.
23:16	<b>HeaderTypeReg</b> . Value: 0.
15:8	<b>LatencyTimer</b> . Value: 0.
7:0	<b>CacheLineSize</b> . Read-write. Reset: 0. This field specifies the system cache line size in units of double words. This field is implemented as a read-write field for legacy compatibility purposes only and has no functional impact.

**D14F2x10 Lower Base Address Register**

Reset: 0000\_0004h.

Bits	Description
31:14	<b>HDABaseAddr[31:14]: low base address</b> . Read-write. Lower Base Address for the HD Audio controller's memory mapped configuration registers.
13:4	Reserved.
3	<b>Pref: prefetchable</b> . Read-only. 0=Non-prefetchable memory region.
2:1	<b>Type: base address register type</b> . Read-only. 10b=64 bit base address register.
0	<b>MemSpace: memory space type</b> . Read-only. 0=Memory mapped base address.

**D14F2x14 Upper Base Address Register**

Reset: 0000\_0000h.

Bits	Description
31:0	<b>HDABaseAddr[63:32]: Upper Base Address</b> . Read-write. Upper Base Address for the HD Audio controller's memory mapped configuration registers.

**D14F2x2C Subsystem and Subvendor ID**

Reset: 0000\_0000h.

Bits	Description
31:16	<b>SubsystemID</b> . Write-once.
15:0	<b>SubsystemVendorID</b> . Write-once.

**D14F2x34 Capabilities Pointer**

Reset: 0000\_0050h.

Bits	Description
31:8	Reserved.
7:0	<b>CapPtr: capabilities pointer</b> . Read-only. Pointer to PM capability.

**D14F2x3C Interrupt Line**

Reset: 0000\_0100h.

Bits	Description
31:24	<b>MaximumLatency.</b> Read-only.
23:16	<b>MinimumGrant.</b> Read-only.
15:12	Reserved.
11:8	<b>InterruptPin: interrupt pin.</b> Read-only. This register reflects the value programs into Interrupt Control Pin register <a href="#">D14F2x44</a> [InterruptPinControl].
7:0	<b>InterruptLine: interrupt line.</b> Read-write. This field contains the interrupt line routing information. This is not used by the HD Audio controller.

**D14F2x40 Misc Control**

Read-write. Reset: 0000\_0000h.

Bits	Description										
31:23	Reserved.										
22	<b>EnableStopALinkClock.</b> 1=Enable stop A-Link clock indication. 0=Disable.										
21	<b>EnableClockGating.</b> 1=Enable clock gating. 0=Disable.										
20	<b>DisableMinimumRetry.</b> 1=Disable minimum retry on ALink Bus. 0=Enable. When enabled, after the first request of a transaction, it will not generate another request for the same transaction unless notified by AB the data is available.										
19	Reserved.										
18	<b>EnableNoSnoopRequest.</b> 1=Enable No Snoop request to ACPI. 0=Disable. No snoop request to ACPI When enabled and the DMA cycle is No Snoop, ACPI will not generate a wake to CPU in C2 state.										
17	<b>DisableNoSnoopOverride.</b> 1=DisableNoSnoop controlling the No Snoop attribute. 0=Override DisableNoSnoop setting and always generate No Snoop attribute on Buffer Descriptor and Data Buffer DMA.										
16	<b>DisableNoSnoop.</b> 1=No Snoop attribute is disabled on Buffer Descriptor and Data Buffer DMA. 0=Set the No Snoop attribute on Buffer Descriptor and Data Buffer DMA when the Traffic Priority bit is set in the Stream Descriptor.										
15:9	Reserved.										
8	<b>StaticOutputFIFOSizeEnable.</b> 1=Enable Static Output FIFO Size. 0=Output FIFO Size is set dynamically set based on Stream Format.										
7:2	Reserved.										
1:0	<b>StaticOutputFIFOSizeSelect.</b> Static Output FIFO Size is functional only when ( <a href="#">D14F2x40</a> [StaticOutputFIFOSizeEnable]==1). <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>1/8 of the maximum Output FIFO Size.</td> </tr> <tr> <td>01b</td> <td>1/4 of the maximum Output FIFO size</td> </tr> <tr> <td>10b</td> <td>1/2 of the maximum Output FIFO Size</td> </tr> <tr> <td>11b</td> <td>Use the maximum Output FIFO Size</td> </tr> </tbody> </table>	Bits	Definition	00b	1/8 of the maximum Output FIFO Size.	01b	1/4 of the maximum Output FIFO size	10b	1/2 of the maximum Output FIFO Size	11b	Use the maximum Output FIFO Size
Bits	Definition										
00b	1/8 of the maximum Output FIFO Size.										
01b	1/4 of the maximum Output FIFO size										
10b	1/2 of the maximum Output FIFO Size										
11b	Use the maximum Output FIFO Size										



**D14F2x44 Interrupt Pin/Capability**

Reset: 0000\_0001h.

Bits	Description
31:16	Reserved.
15:11	Reserved.
10	IF (RevA0) THEN Reserved. ELSE <b>BClkOnInD3</b> . Read-write. Reset: 0. 1=Do not turn off AZ_BCLK in D3. ENDIF.
9	<b>EnAFCap</b> . Read-write. BIOS: 0. 1=Enable Advanced Features Capability. See <a href="#">Table 82</a> .
8	<b>EnMsiCap</b> . Read-write. BIOS: 1. 1=Enable MSI Capability. See <a href="#">Table 82</a> .
7:4	Reserved.
3:0	<b>InterruptPinControl</b> . Read-write. Controls the value reports in <a href="#">D14F2x3C</a> [InterruptPin].

**D14F2x50 Power Management Capability**

Bits	Description
31:27	<b>PmeSupport</b> . Value: 11001b. Indicates PME# can be generated from D0 and D3 states.
26	<b>D2Support: D2 support</b> . Value: 0. Indicates that D2 is not supported in hardware.
25	<b>D1Support: D1 support</b> . Value: 0. Indicates that D1 is not supported in hardware.
24:22	<b>AuxCurrent: auxiliary current</b> . Value: 001b. Indicates 55mA maximum suspend well current is required in the D3cold state.
21	<b>DevSpecificInit: device specific initialization</b> . Value: 0. Indicates that there is no device specific initialization necessary.
20	Reserved.
19	<b>PmeClock</b> . Value: 0. Indicates that no PCI clock is required for the function to generate PME#.
18:16	<b>Version: version</b> . Value: 010b. Indicates this function complies with Revision 1.1 of the PCI Power Management Interface Specification
15:8	<b>NextPtr: next pointer</b> . IF ( <a href="#">D14F2x44</a> [EnAFCap] && <a href="#">D14F2x44</a> [EnMsiCap]) THEN Value: 60h. ELSEIF ( <a href="#">D14F2x44</a> [EnAFCap]) THEN Value: 70h. ELSEIF ( <a href="#">D14F2x44</a> [EnMsiCap]) THEN Value: 60h. ELSE Value: 00h. ENDIF. Specifies the address of the next capability structure, or zero if this the end of the linked list of capability structures. See <a href="#">Table 82</a> .
7:0	<b>CapID: capability ID</b> . Value: 01h. Indicates that the capability structure is a PCI power management data structure.

**D14F2x54 Power Management Control and Status**

Bits	Description												
31:16	Reserved.												
15	<b>PmeStatus: PME status.</b> Read; set-by-hardware; write-1-to-clear. Cold reset: 0. This bit set when the HD Audio controller asserts the PME# signal, and is independent of the PME Enable bit.												
14:9	Reserved.												
8	<b>PmeEn: PME# enable.</b> Read-write. Cold reset: 0. Enables the function to assert PME#.												
7:2	Reserved.												
1:0	<b>PowerState: power state.</b> Read-write. Reset: 0. Specifies the current power state of the root port and sets the root port into a new power state. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Bits</th> <th>Definition</th> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>D0</td> <td>10b</td> <td>Reserved.</td> </tr> <tr> <td>01b</td> <td>Reserved</td> <td>11b</td> <td>D3hot</td> </tr> </tbody> </table>	Bits	Definition	Bits	Definition	00b	D0	10b	Reserved.	01b	Reserved	11b	D3hot
Bits	Definition	Bits	Definition										
00b	D0	10b	Reserved.										
01b	Reserved	11b	D3hot										

**D14F2x60 MSI Capability**

Bits	Description
31:24	Reserved.
23	<b>Msi64bit: MSI 64 bit capability.</b> Read-only. Reset: 1. 1=The device is capable of sending 64-bit MSI messages.
22:20	<b>MsiMultiEn: MSI multiple message enable.</b> 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	<b>MsiMultiCap: MSI multiple message capability.</b> Read-only. Reset: 000b. 000b=The device is requesting one vector.
16	<b>MsiEn: MSI enable.</b> 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	<b>NextPtr: next pointer.</b> IF (D14F2x44[EnAFCap]) THEN Value: 70h. ELSE Value: 00h. ENDIF. Specifies the address of the next capability structure, or zero if this the end of the linked list of capability structures. See <a href="#">Table 82</a> .
7:0	<b>CapID: capability ID.</b> IF (D14F2x44[EnMsiCap]==1) THEN Value: 05h. ELSE Value: 00h. ENDIF. 05h=MSI capability structure. See <a href="#">Table 82</a> .

**D14F2x64 MSI Message Address Low**

Reset: 0000\_0000h.

Bits	Description
31:2	<b>MsiMsgAddrLo: MSI message address.</b> Read-write. This register specifies the doubleword aligned address for the MSI memory write transaction.
1:0	Reserved.

**D14F2x68 MSI Message Address High**

Reset: 0000\_0000h.

Bits	Description
31:8	Reserved.
7:0	<b>MsiMsgAddrHi: MSI message address.</b> Read-write. This register specifies the upper 8 bits of the MSI address in 64-bit MSI mode.

**D14F2x6C MSI Message Data**

Reset: 0000\_0000h.

Bits	Description
31:16	Reserved.
15:0	<b>MsiData: MSI message data.</b> Read-write. This register specifies lower 16 bits of data for the MSI memory write transaction. The upper 16 bits are always 0.

**D14F2x70 Advanced Features Capability**

Bits	Description
31:26	Reserved.
25	<b>TPCapabilities.</b> Read-only. IF (D14F2x44[EnAFCap]==1) THEN Reset: 1. ELSE Reset: 0. ENDIF. 1=Support for Transactions Pending. See Table 82.
24	<b>FLRCapabilities.</b> Read-only. IF (D14F2x44[EnAFCap]==1) THEN Reset: 1. ELSE Reset: 0. ENDIF. 1=Support for Function Level Reset (FLR). See Table 82.
23:16	<b>AFLength.</b> Read-only. IF (D14F2x44[EnAFCap]==1) THEN Reset: 06h. ELSE Reset: 00h. ENDIF. Indicates AF Structure Length (bytes). See Table 82.
15:8	<b>NextPtr: next pointer.</b> Read-only. Reset: 00h. 00h=This the end of the linked list of capability structures. See Table 82.
7:0	<b>AFCapID: capability ID.</b> Read-only. IF (D14F2x44[EnAFCap]==1) THEN Reset: 13h. ELSE Reset: 00h. ENDIF. 13h=Indicates Advanced Features Capability. See Table 82.

**D14F2x74 Advanced Features Control and Status**

Bits	Description
31:9	Reserved.
8	<b>TransactionPending.</b> Read-only; updated-by-hardware. Reset: 0. 1=Indicates that the function has issued one or more non-posted transactions which have not been completed including non-posted transactions that a target has terminated with retry. 0=Indicates that all non-posted transactions have been completed.

7:1	Reserved.
0	<b>InitiateFLR.</b> RAZ; write-1-only; cleared-when-done. Reset: 0. A write of 1 initiates Function Level Reset. FLR requirements are defined in the PCI Express® Base Specification. Registers and state information that do not apply to conventional PCI are exempt from the FLR requirements given there.

### 3.25.6 Secure Digital (SD) Controller

#### 3.25.6.1 Device 14h Function 7 Configuration Registers (SD)

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.

##### D14F7x00 Device/Vendor ID

Bits	Description
31:16	<b>DeviceID: Device ID.</b> Value: 7813h.
15:0	<b>VendorID: Vendor ID.</b> Value: 1022h. Specifies a unique 16-bit value assigned to a vendor.

##### D14F7x04 Status/Command

Bits	Description
31	<b>DetectedParityError: Detected Parity Error.</b> Read; updated-by-hardware; write-1-to-clear. Reset: 0. 1=The FCH detects a parity error.
30	<b>SignaledSystemError: Signaled System Error.</b> Read; updated-by-hardware; write-1-to-clear. Reset: 0. SERR# status. 1=The FCH detects a PCI address parity error.
29	<b>ReceivedMasterAbort: Received Master Abort.</b> Read; updated-by-hardware; write-1-to-clear. Reset: 0. 1=The FCH acts as a PCI master and aborts a PCI bus memory cycle.
28	<b>ReceivedTargetAbort: Received Target Abort.</b> Read; updated-by-hardware; write-1-to-clear. Reset: 0. 1=FCH received Target Abort from a PCI target.
27	<b>SignaledTargetAbort: Signaled Target Abort.</b> Read; updated-by-hardware; write-1-to-clear. Reset: 0. 1=The FCH signaled Target Abort while acting as Bus Master.
26:25	<b>DeviceSelectTiming: Device Select Timing.</b> Value: 01b. 01b=DEVSEL# Medium timing when performing a positive decode.
24	<b>MasterDataParityError: Master Data Parity Error.</b> Read; updated-by-hardware; write-1-to-clear. Reset: 0. Data Parity reported. 1=The FCH detects PERR# asserted while acting as PCI master.
23:21	Reserved.
20	<b>CapabilitiesList: Capabilities List.</b> Value: 1.
19:10	Reserved.
9	<b>FastBack2BackEnable: Fast Back-to-Back Enable.</b> Value: 0.
8	<b>SERREnable: SERR# Enable.</b> Read-write. Reset: 0. 1=FCH asserts SERR# when it detects as address parity error.
7	<b>SteppingControl: Stepping Control.</b> Value: 0. 1=Wait Cycle enable.
6	<b>ParityErrorResponse: Parity Error Response.</b> Read-write. Reset: 0. PERR# (Response) detection enable bit. 1=The FCH asserts PERR# when it is the agent receiving data and it detects a parity error. 0=PERR# is not asserted.
5	<b>VGAPaletteSnoop: VGA Palette Snoop.</b> Value: 0. 1=VGA Palette Snoop Enable.
4	<b>MemoryWriteandInvalidateEnable: Memory Write and Invalidate Enable.</b> Value: 0. 1=Memory write and invalidate enable.
3	<b>SpecialCycles: Special Cycles.</b> Value: 0. 1=Special cycle recognition enable.

2	<b>BusMaster: Bus Master.</b> Read-write. Reset: 0. 1=Bus master function enable.
1	<b>MemorySpace: Memory Space.</b> Read-write. Reset: 0. 1=Enable memory access. 0=Disable.
0	<b>IOSpace: IO Space.</b> Value: 0. 1=IO access enable.

**D14F7x08 Revision ID/Class Code**

Bits	Description										
31:24	<b>ClassCode: Class Code.</b> Read-only. Reset: 8h. Basic Class.										
23:16	<b>SubClass: Sub Class.</b> Read-only. Reset: 5h. For SD host controller										
15:8	<b>InterfaceCode: Interface Code.</b> Value: 1. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00h</td> <td>Standard host not supporting DMA</td> </tr> <tr> <td>01h</td> <td>Standard host supporting DMA</td> </tr> <tr> <td>02h</td> <td>Vendor unique SD host controller</td> </tr> <tr> <td>FFh-03h</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Definition	00h	Standard host not supporting DMA	01h	Standard host supporting DMA	02h	Vendor unique SD host controller	FFh-03h	Reserved
Bits	Definition										
00h	Standard host not supporting DMA										
01h	Standard host supporting DMA										
02h	Vendor unique SD host controller										
FFh-03h	Reserved										
7:0	<b>RevisionID: Revision ID.</b> Value: 01h.										

**D14F7x0C Cache Line Size**

Bits	Description
31:24	<b>Bist.</b> Read-only. Reset: 0. No BIST modes.
23:16	<b>HeaderType: Header Type.</b> Read-only. Reset: 80h. Specifies the type of the predefined header in the configuration space.
15:8	<b>LatencyTimer: Latency Timer.</b> Read-only. Reset: 0. Specifies the value of the Latency Timer in units of PCICLKs.
7:0	<b>CacheLineSize: Cache Line Size.</b> Read-only. Reset: 0.

**D14F7x10 Base Address Reg 0**

Bits	Description
31:8	<b>BaseAddress0: Base Address 0.</b> Read-write. Reset: 0h. Specifies base address 0[31:8].
7:4	Reserved.
3	<b>Prefetchable.</b> Value: 0. 0=Non-prefetchable.
2:1	<b>Type.</b> Value: 10b. 00b=32-bit base address. 10b=64-bit base address.
0	<b>SpaceIndicator: Space Indicator.</b> Value: 0. 0=Memory-mapped base address.

**D14F7x14 Upper Base Address Reg 0**

Bits	Description
31:0	<b>BaseAddress0Upper: Base Address 0 Upper.</b> Read-write. Reset: 0. Specifies base address 0[63:32].

**D14F7x2C Subsystem ID and Subsystem Vendor ID**

Write-once. This 4-byte register is a write-once & read-only afterward register. The BIOS writes this register once (all 4 bytes at once) and software reads its value (when needed).

Bits	Description
31:16	<b>SubsystemID: Subsystem ID.</b> Reset: 7813h. Subsystem ID.
15:0	<b>SubsystemVendorID: Subsystem Vendor ID.</b> Reset: 1022h. Subsystem Vendor ID.

**D14F7x34 Capabilities Pointer**

Bits	Description
31:8	Reserved.
7:0	<b>CapabilitiesPointer: Capabilities Pointer.</b>

**D14F7x3C Interrupt Line**

Bits	Description
31:16	Reserved.
15:8	<b>InterruptPin: Interrupt Pin.</b> Read-only. Reset: 1. Interrupt pin.
7:0	<b>InterruptLine: Interrupt Line.</b> Read-write. Reset: 0. Interrupt line.

**D14F7x40 Slot Information**

Bits	Description																				
31:7	Reserved.																				
6:4	<p><b>NumberOfSlots: Number of Slots.</b> Read-write. Reset: 0. Number of slots supported. These bits indicate the number of slots the Host Controller supports. In the case of a single function, a maximum of 6 slots can be assigned.</p> <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>1 slot</td> <td>100b</td> <td>5 slots</td> </tr> <tr> <td>001b</td> <td>2 slots</td> <td>101b</td> <td>6 slots</td> </tr> <tr> <td>010b</td> <td>3 slots</td> <td>110b</td> <td>Reserved</td> </tr> <tr> <td>011b</td> <td>4 slots</td> <td>111b</td> <td>Reserved</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	<u>Bits</u>	<u>Definition</u>	000b	1 slot	100b	5 slots	001b	2 slots	101b	6 slots	010b	3 slots	110b	Reserved	011b	4 slots	111b	Reserved
<u>Bits</u>	<u>Definition</u>	<u>Bits</u>	<u>Definition</u>																		
000b	1 slot	100b	5 slots																		
001b	2 slots	101b	6 slots																		
010b	3 slots	110b	Reserved																		
011b	4 slots	111b	Reserved																		

3	Reserved.																				
2:0	<b>FirstBaseAddressRegisterNumber: First Base Address Register Number.</b> Read-write. Reset: 0. Up to 6 base addresses can be specified in single configuration. These bits indicate the first base address register number assigned to SD Host Controller register set. In the case of single function and multiple register sets, contiguous base addresses are used. <b>D14F7x40[NumberOfSlots]</b> specifies the number of base addresses.																				
	<table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>Base Address 10h (BAR0)</td> <td>100b</td> <td>Base Address 20h (BAR4)</td> </tr> <tr> <td>001b</td> <td>Base Address 14h (BAR1)</td> <td>101b</td> <td>Base Address 24h (BAR5)</td> </tr> <tr> <td>010b</td> <td>Base Address 18h (BAR2)</td> <td>110b</td> <td>Reserved</td> </tr> <tr> <td>011b</td> <td>Base Address 1Ch (BAR3)</td> <td>111b</td> <td>Reserved</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	<u>Bits</u>	<u>Definition</u>	000b	Base Address 10h (BAR0)	100b	Base Address 20h (BAR4)	001b	Base Address 14h (BAR1)	101b	Base Address 24h (BAR5)	010b	Base Address 18h (BAR2)	110b	Reserved	011b	Base Address 1Ch (BAR3)	111b	Reserved
<u>Bits</u>	<u>Definition</u>	<u>Bits</u>	<u>Definition</u>																		
000b	Base Address 10h (BAR0)	100b	Base Address 20h (BAR4)																		
001b	Base Address 14h (BAR1)	101b	Base Address 24h (BAR5)																		
010b	Base Address 18h (BAR2)	110b	Reserved																		
011b	Base Address 1Ch (BAR3)	111b	Reserved																		

**D14F7x80 SD PCI MSI Capability Header**

Bits	Description
31:24	Reserved.
23	<b>ExtendAddrEn.</b> Value: 1. 64-bit Address Capable.
22:20	<b>MulMsgEn.</b> Read-write. Reset: 0. Multiple Message Enable.
19:17	<b>MulMsgCap.</b> Read-only. Reset: 0. Multiple Message Capable.
16	<b>MsiEnable.</b> Read-write. Reset: 0. MSI Enable.
15:8	<b>CapNxtPtr.</b> Value: 0. Reset: 0.
7:0	<b>CapId.</b> Read-only. Reset: 5h. A value of 05h indicates MSI.

**D14F7x84 SD PCI MSI Address**

Bits	Description
31:2	<b>MsgAddr.</b> Read-write. Reset: 0. Specifies Message Address[31:2].
1:0	Reserved.

**D14F7x88 SD PCI MSI Upper Address**

Bits	Description
31:0	<b>MsgUpperAddr.</b> Read-write. Reset: 0. Specifies Message Upper Address[63:32].

**D14F7x8C SD PCI MSI Data**

Bits	Description
31:16	Reserved.
15:0	<b>MsgData.</b> Read-write. Reset: 0. Message Data.



**D14F7x90 Power Management Capability Header**

Bits	Description												
31:27	<b>PMESupport: PME Support.</b> Read-only. Reset: 0. This 5-bit field indicates the power states in which the function may assert PME#. A value of 0b for any bit indicates that the function is not capable of asserting the PME# signal while in that power state. <table border="0"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>xxxx1b</td> <td>PME# can be asserted from D0</td> </tr> <tr> <td>xxx1xb</td> <td>PME# can be asserted from D1</td> </tr> <tr> <td>xx1xxb</td> <td>PME# can be asserted from D2</td> </tr> <tr> <td>x1xxxb</td> <td>PME# can be asserted from D3hot</td> </tr> <tr> <td>1xxxxb</td> <td>PME# can be asserted from D3cold</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	xxxx1b	PME# can be asserted from D0	xxx1xb	PME# can be asserted from D1	xx1xxb	PME# can be asserted from D2	x1xxxb	PME# can be asserted from D3hot	1xxxxb	PME# can be asserted from D3cold
<u>Bits</u>	<u>Definition</u>												
xxxx1b	PME# can be asserted from D0												
xxx1xb	PME# can be asserted from D1												
xx1xxb	PME# can be asserted from D2												
x1xxxb	PME# can be asserted from D3hot												
1xxxxb	PME# can be asserted from D3cold												
26	<b>D2Support: D2 Support.</b> Read-only. Reset: 0. 1=This function supports the D2 Power Management State.												
25	<b>D1Support: D1 Support.</b> Read-only. Reset: 0. 1=This function supports the D1 Power Management State.												
24:22	<b>AuxCurrent: Aux Current.</b> Read-only. Reset: 0. This 3 bit field reports the 3.3V auxiliary current requirements for the PCI function.												
21	<b>Dsi.</b> Read-only. Reset: 0. The Device Specific Initialization required. 1=Indicates special initialization of this function is required (beyond the standard PCI configuration header) before the generic class device driver is able to use it.												
20	Reserved.												
19	<b>PmeClock: PME Clock.</b> Value: 0. PME Clock. 0=Indicates that no PCI clock is required for the function to generate PME#. 1=Indicates that the function relies on the presence of the PCI clock for PME# operation.												
18:16	<b>Version.</b> Value: 011b. 011b=Indicates that this function complies with revision 1.2 of the PCI Power Management Interface Specification.												
15:8	<b>CAPNextPointer: CAP Next Pointer.</b> Read-only. Reset: 0. No other capability.												
7:0	<b>CAPID: CAP ID.</b> Read-only. Reset: 1. Power management.												

**D14F7x94 Power Management Control and Status Register**

Bits	Description
31:24	<b>Data.</b> Read-only. Reset: 0. Reports the state dependent data requested by the DataSelect field. The value of this register is scaled by the value reported by the DataScale field.
23	<b>BPCCEn: Bus Power/Clock Control Enable.</b> Read-only. Reset: 0. Bus Power/Clock Control Enable.
22	<b>B2B3: B2 B3#.</b> Read-only. Reset: 0. B2/B3 support for D3hot.
21:16	Reserved.
15	<b>PMEStatus: PME Status.</b> Read-only; updated-by-hardware. Reset: 0. Set when the function normally asserts the PME# signal independent of the state of PMEEEn.
14:13	<b>DataScale: Data Scale.</b> Read-only. Reset: 0. Indicates the scaling factor to be used when interpreting the value of the Data register. The value and meaning of this field will vary depending on which data value has been selected by the DataSelect field.

12:9	<b>DataSelect: Data Select.</b> Read-only. Reset: 0. Selects which data is to be reported through the Data register and DataScale field.												
8	<b>PMEEEn: PME En.</b> Read-only. Reset: 0. 1=Enable the function to assert PME#. 0=PME# assertion is disabled.												
7:4	Reserved.												
3	<b>NoSoftReset: No Soft Reset.</b> Read-only. Reset: 0. 1=Indicates that devices transitioning from D3hot to D0 because of PowerState commands do not perform an internal reset.												
2	Reserved.												
1:0	<p><b>Power.</b> Read-only. Reset: 0. Determines the current power state of a function and to set the function into a new power state.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>D0</td> <td>10b</td> <td>D2</td> </tr> <tr> <td>01b</td> <td>D1</td> <td>11b</td> <td>D3hot</td> </tr> </tbody> </table> <p>If software attempts to write an unsupported, optional state to this field, the write operation must complete normally on the bus; however, the data is discarded and no state change occurs.</p>	Bits	Definition	Bits	Definition	00b	D0	10b	D2	01b	D1	11b	D3hot
Bits	Definition	Bits	Definition										
00b	D0	10b	D2										
01b	D1	11b	D3hot										

### D14F7xB8 SD Auto Pattern

Bits	Description
31:28	<b>SdDrvStr33Sn.</b> Read-write. Reset: 0. Specifies the SN driver strength value for 3.3V.
27:24	<b>SdDrvStr33Sp.</b> Read-write. Reset: 0. Specifies the SP driver strength value for 3.3V.
23:0	Reserved.

### 3.25.6.2 SD Host Controller Configuration Registers (SDHC)

The registers are accessed directly mapped by [D14F7x10 \[Base Address Reg 0\]](#) and [D14F7x14 \[Upper Base Address Reg 0\]](#).

#### SDHCx00 SDHC System Address / Argument 2

Bits	Description
31:16	<b>SysAddr1.</b> Read-write. Reset: 0. System Address upper bits. Updating this register clears DMA_WAIT. It indicates system memory address for DMA. When DMA transfer detects the DMA Buffer Boundary specified by the Host DMA Buffer Boundary in the Block Size register, SD controller asserts DMA_WAIT. Also SD controller generates DMA interrupt at this time when corresponding bits in the Normal Interrupt Status Enable register and Normal Interrupt Signal Enable register are set. While ADMA is enabled, this register will not be used.
15:0	<b>SysAddr0.</b> Read-write. Reset: 0. System Address lower bits. Updating this register clears DMA_WAIT.

#### SDHCx04 SDHC Block CS

Block Size and Block Count.

Bits	Description																				
31:16	<b>BlkCnt.</b> Read-write. Reset: 0. Block Count. It indicates block count of multiple data transfer. It is enabled when the Block Count Enable bit (D01) in the Transfer Mode register is set to 1. It is decremented after each block data transmission. During infinite data transmission, setting of this bit is meaningless.																				
15	Reserved.																				
14:12	<b>DmaBufBndry.</b> Read-write. Reset: 0. Host DMA Buffer Boundary. Indicates the contiguous buffer size in the system memory. When the boundary is reached, DMA interrupt will be generated. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>4K bytes</td> <td>100b</td> <td>64K bytes</td> </tr> <tr> <td>001b</td> <td>8K bytes</td> <td>101b</td> <td>128K bytes</td> </tr> <tr> <td>010b</td> <td>16K bytes</td> <td>110b</td> <td>256K bytes</td> </tr> <tr> <td>011b</td> <td>32K bytes</td> <td>111b</td> <td>512K bytes</td> </tr> </tbody> </table>	Bits	Definition	Bits	Definition	000b	4K bytes	100b	64K bytes	001b	8K bytes	101b	128K bytes	010b	16K bytes	110b	256K bytes	011b	32K bytes	111b	512K bytes
Bits	Definition	Bits	Definition																		
000b	4K bytes	100b	64K bytes																		
001b	8K bytes	101b	128K bytes																		
010b	16K bytes	110b	256K bytes																		
011b	32K bytes	111b	512K bytes																		
11:0	<b>BlkSize.</b> Read-write. Reset: 0. Transfer Data Length (max. block size is 2K bytes). When the CE-ATA Enable bit of the CE-ATA Control register is set, a value of 0x000 indicates block size of 4K bytes.																				

#### SDHCx08 SDHC Command Argument

Bits	Description
31:16	<b>ARGUMENT1.</b> Read-write. Reset: 0. Upper bits. Command Argument. Command arguments specified as bits [39:8] of the command format.
15:0	<b>ARGUMENT0.</b> Read-write. Reset: 0. Lower bits.

#### SDHCx0C SDHC Command/Transfer Mode

Bits	Description										
31:30	Reserved.										
29:24	<b>CmdIdx.</b> Read-write. Reset: 0. Command Index.										
23:22	<b>CmdType.</b> Read-write. Reset: 0. Command Type. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Normal.</td> </tr> <tr> <td>01b</td> <td>Suspend CMD52 for writing BR in CCCR.</td> </tr> <tr> <td>10b</td> <td>Resume CMD52 for writing Function Sel in CCCR.</td> </tr> <tr> <td>11b</td> <td>Abort CMD12 (SD Memory) or Abort CMD52 (SDIO).</td> </tr> </tbody> </table>	Bits	Definition	00b	Normal.	01b	Suspend CMD52 for writing BR in CCCR.	10b	Resume CMD52 for writing Function Sel in CCCR.	11b	Abort CMD12 (SD Memory) or Abort CMD52 (SDIO).
Bits	Definition										
00b	Normal.										
01b	Suspend CMD52 for writing BR in CCCR.										
10b	Resume CMD52 for writing Function Sel in CCCR.										
11b	Abort CMD12 (SD Memory) or Abort CMD52 (SDIO).										
21	<b>DataPrsnt.</b> Read-write. Reset: 0. Data Present Select. Indicates data is present and will be transferred on the DAT line. When command is issued with this bit enabled, internal buffer will be cleared. 0=No data. 1=Data.										
20	<b>CmdIdxChkEn.</b> Read-write. Reset: 0. 1=Command Index Check Enable. 0=Disable.										
19	<b>CrcChkEn.</b> Read-write. Reset: 0. 1=Command CRC Check Enable. 0=Disable.										
18	Reserved.										

17:16	<b>RespType.</b> Read-write. Reset: 0. Response Type Select. <u>Bits</u> <u>Definition</u> 00b                          No Response. 01b                          Response length is 136 bits. 10b                          Response length is 48 bits without busy. 11b                          Response length is 48 bits with busy.
15:6	Reserved.
5	<b>MultiBlk.</b> Read-write. Reset: 0. 1=Multiple/Single Block Select. 0=Single Block.
4	<b>DataDir.</b> Read-write. Reset: 0. Data Transfer Direction. 0=Write. 1=Read.
3:2	<b>AutoCmdEn.</b> Read-write. Reset: 0. <u>Bits</u> <u>Definition</u> 00b                          Auto Command Disabled 01b                          Auto CMD12 Enable 10b                          Auto CMD23 Enable 11b                          Reserved
1	<b>BlkCntEn.</b> Read-write. Reset: 0. 1=Block Count Enable.
0	<b>DmaEn.</b> Read-write. Reset: 0. 1=DMA Enable. 0=Disable.

**SDHCx10 SDHC\_RESP1\_0**

Bits	Description
31:16	<b>RESPONSE1.</b> Read-only. Reset: 0. R39-24 of response is saved in this register.
15:0	<b>RESPONSE0.</b> Read-only. Reset: 0. R23-8 of response is saved in this register.

**SDHCx14 SDHC\_RESP3\_2**

Bits	Description
31:16	<b>RESPONSE3.</b> Read-only. Reset: 0. R71-56 of response is saved in this register.
15:0	<b>RESPONSE2.</b> Read-only. Reset: 0. R55-40 of response is saved in this register.

**SDHCx18 SDHC\_RESP5\_4**

Bits	Description
31:16	<b>RESPONSE5.</b> Read-only. Reset: 0. R103-88 of response is saved in this register.
15:0	<b>RESPONSE4.</b> Read-only. Reset: 0. R87-72 of response is saved in this register.

**SDHCx1C SDHC\_RESP7\_6**

Bits	Description
------	-------------

31:16	<b>RESPONSE7.</b> Read-only. Reset: 0. R127-120 of response or R39-24 of Auto CMD12 response is saved in this register.
15:0	<b>RESPONSE6.</b> Read-only. Reset: 0. R119-104 of response or R23-8 of Auto CMD12 response is saved in this register.

**SDHCx20 SDHC\_BUFFER**

Bits	Description
31:16	<b>BuffData1.</b> Read-write. Reset: 0. Upper bits. Data Buffer. Data will be accessed through this register. Data which exceeds the size designated by the Block Size register will not be written in the data buffer.
15:0	<b>BuffData0.</b> Read-write. Reset: 0. Lower bits.

**SDHCx24 SDHC\_PRNT\_STATE**

Bits	Description
31:25	Reserved.
24	<b>CmdLevel.</b> Read-only. Reset: 0. CMD Line Signal Level. Reflects signal level of CMD line.
23:20	<b>DatLevel.</b> Read-only. Reset: 0. DAT Line Signal Level. Reflects signal level of DAT line.
19	<b>WpLevel.</b> Read-only. Reset: 0. Write Protect Switch Level. 0=Write protected. 1=Write enable.
18	<b>CdLevel.</b> Read-only. Reset: 0. Card Detect Pin Level. 0=No card present. 1=Card present.
17	<b>CardStable.</b> Read-only. Reset: 0. Card State Stable. Indicates Card Detect signal level is stable. 0=Not stable (debouncing or resetting). 1=Card stable.
16	<b>CardIns.</b> Read-only. Reset: 0. Card Inserted. 0=No card inserted or debouncing state or resetting. 1=Card inserted.
15:12	Reserved.
11	<b>BufRdEn.</b> Read-only. Reset: 0. Buffer Read Enable. Indicates buffer is ready for reading. 0=Read disable. 1=Read enable.
10	<b>BufWrEn.</b> Read-only. Reset: 0. Buffer Write Enable. Indicates buffer is ready for writing. 0=Write disable. 1=Write enable.
9	<b>RdTxActive.</b> Read-only. Reset: 0. Read Transfer Active. Indicates occurrence of read data transfer. 0=No data transferring. 1=Read data transferring.
8	<b>WrTxActive.</b> Read-only. Reset: 0. Write Transfer Active. Indicates occurrence of write data transfer. 0=No data transferring. 1=Write data transferring.
7:3	Reserved.
2	<b>DatLineActive.</b> Read-only. Reset: 0. DAT Line Active. Indicates DAT line on SD Bus is active. 0=DAT line inactive. 1=DAT line active.
1	<b>CmdInhibDat.</b> Read-only. Reset: 0. Command Inhibit (DAT). Indicates that commands which use also the DAT line can be issued. 0=Can issue commands which use DAT line. 1=Cannot issue any commands which use DAT line.
0	<b>CmdInhibCmd.</b> Read-only. Reset: 0. Command Inhibit (CMD). Indicates that commands which use only the CMD line can be issued. 0=Can issue commands which use CMD line. 1=Cannot issue any commands.

**SDHCx28 SDHC\_CTRL1**

Bits	Description												
31:27	Reserved.												
26	<b>SdRemWakeEn.</b> Read-write. Reset: 0. 1=SD Card Removal Wake up. 0=Disable.												
25	<b>SdInsWakeEn.</b> Read-write. Reset: 0. 1=SD Card Insertion Wake up. 0=Disable.												
24	<b>SdIntWakeEn.</b> Read-write. Reset: 0. 1=SD Card Interrupt Wake up. 0=Disable.												
23:20	Reserved.												
19	<b>BgIntEn.</b> Read-write. Reset: 0. Interrupt at Block Gap. 1=Enable interrupt detection during 4-bit block transmission. 0=Disable.												
18	<b>ReadWaitEn.</b> Read-write. Reset: 0. Read Wait Control. 1=Indicates Read Wait will be inserted when needed. 0=Disable.												
17	<b>ContReq.</b> Read-write. Reset: 0. Continue Request. Writing 1 to this bit triggers restart of halted data transaction with current register setting. Once this bit is 1, the internal buffer will be cleared and data transfer sequence will be restarted. 1=Restart.												
16	<b>BgStopReq.</b> Read-write. Reset: 0. Stop at Block Gap Request. Writing 1 to this bit triggers halting of current data transfer after next block gap. To use this request, the Read Wait function is necessary in read transaction. Even if the Auto CMD12 Enable bit is set to 1, Auto CMD12 is not issued in case this bit is set to 1. This bit is cleared by not only writing 0 to this bit, but also issuing abort commands. 0=Transfer. 1=Stop.												
15:12	Reserved.												
11:9	<b>SdBusVoltage.</b> Read-write. Reset: 0. SD Bus Voltage. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;"><u>Bits</u></th> <th style="text-align: left;"><u>Definition</u></th> <th style="text-align: left;"><u>Bits</u></th> <th style="text-align: left;"><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>100b-000b</td> <td>Reserved</td> <td>110b</td> <td>3.0 Volts</td> </tr> <tr> <td>101b</td> <td>1.8 Volts</td> <td>111b</td> <td>3.3 Volts</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	<u>Bits</u>	<u>Definition</u>	100b-000b	Reserved	110b	3.0 Volts	101b	1.8 Volts	111b	3.3 Volts
<u>Bits</u>	<u>Definition</u>	<u>Bits</u>	<u>Definition</u>										
100b-000b	Reserved	110b	3.0 Volts										
101b	1.8 Volts	111b	3.3 Volts										
8	<b>SdBusEn.</b> Read-write. Reset: 0. SD Bus Power. Automatically cleared when card is removed. 0=Off. 1=On.												
7	<b>CdTestEn.</b> Read-write. Reset: 0. Card Detect Signal Selection. 0=I/O pin. 1=SD_TEST_LEVEL.												
6	<b>CdTestLevel.</b> Read-write. Reset: 0. Card Detect Test Level. 0=Card removed. 1=Card inserted.												
5	<b>MmcWidth.</b> Read-write. Reset: 0. Extended Data Transfer Width (MMC). 0=Use width set by DAT_TX_WIDTH. 1=Force 8-bit.												
4:3	<b>DmaSelect.</b> Read-write. Reset: 0. DMA Select. Valid only when DMA is enabled. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;"><u>Bits</u></th> <th style="text-align: left;"><u>Definition</u></th> <th style="text-align: left;"><u>Bits</u></th> <th style="text-align: left;"><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>No DMA or SDMA selected</td> <td>10b</td> <td>32-bit ADMA2</td> </tr> <tr> <td>01b</td> <td>32-bit ADMA</td> <td>11b</td> <td>64-bit ADMA2</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	<u>Bits</u>	<u>Definition</u>	00b	No DMA or SDMA selected	10b	32-bit ADMA2	01b	32-bit ADMA	11b	64-bit ADMA2
<u>Bits</u>	<u>Definition</u>	<u>Bits</u>	<u>Definition</u>										
00b	No DMA or SDMA selected	10b	32-bit ADMA2										
01b	32-bit ADMA	11b	64-bit ADMA2										
2	<b>HighSpeedEn.</b> Read-write. Reset: 0. High Speed Enable. When disabled, SD controller outputs commands and data on the falling edge of the SD clock. (Up to 25 MHz SD clock can be supported.) When enabled, SD controller outputs commands and data on the rising edge of the SD clock. (Up to 50 MHz SD clock can be supported.) 0=Normal speed. 1=High speed.												
1	<b>DatTxWidth.</b> Read-write. Reset: 0. Data Transfer Width. 0=1-bit. 1=4-bit.												
0	<b>LedCtrl.</b> Read-write. Reset: 0. LED control. 1=Drives the LED_ON output. 0=Off.												

**SDHCx2C SDHC\_CTRL2**

Bits	Description																														
31:27	Reserved.																														
26	<b>SoftRstDat.</b> Read-write. Reset: 0. Software Reset for DAT Line. The following registers will be cleared: <a href="#">SDHCx20</a> [BuffData1,BuffData0], <a href="#">SDHCx24</a> [BufRdEn, BufWrEn, RdTxActive, WrTxActive, DatLineActive, CmdInhibDat], <a href="#">SDHCx28</a> [ContReq, BgStopReq], <a href="#">SDHCx30</a> [BufRdRdy, BufWrRdy, BlockGapEvt, DatDone].																														
25	<b>SoftRstCmd.</b> Read-write. Reset: 0. Software Reset for CMD Line. The following registers will be cleared: <a href="#">SDHCx24</a> [CmdInhibCmd], <a href="#">SDHCx30</a> [CmdDone].																														
24	<b>SoftRstAll.</b> Read-write. Reset: 0. Software Reset for All. The following registers will not be cleared:- CMD Line Signal Level- DAT[3:0] Line Signal Level- Write Protect Switch Pin Level- Card Detect Pin Level- Card State Stable- Card Inserted- all bits in the Capabilities Register- all bits in the Maximum Current Capabilities Register.																														
23:20	Reserved.																														
19:16	<b>DataToCnt.</b> Read-write. Reset: 0. Data Timeout Counter Value. By using this counter value, DAT line timeouts are detected. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>Eh-0h</td> <td>2^(DataToCnt+13)</td> <td>Fh</td> <td>Reserved.</td> </tr> </tbody> </table>	Bits	Definition	Bits	Definition	Eh-0h	2^(DataToCnt+13)	Fh	Reserved.																						
Bits	Definition	Bits	Definition																												
Eh-0h	2^(DataToCnt+13)	Fh	Reserved.																												
15:8	<b>SdclkDiv[7:0].</b> Read-write. Reset: 0. SDCLK Frequency Select. If multiple bits are set, the most significant bit will be selected. Host Controller Version 1.00 and 2.00: <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>0000_0000b</td> <td>Divide by 1.</td> <td>0001_xxxxb</td> <td>Divide by 32</td> </tr> <tr> <td>0000_0001b</td> <td>Divide by 2.</td> <td>001x_xxxxb</td> <td>Divide by 64.</td> </tr> <tr> <td>0000_001xb</td> <td>Divide by 4.</td> <td>01xx_xxxxb</td> <td>Divide by 128.</td> </tr> <tr> <td>0000_01xxb</td> <td>Divide by 8.</td> <td>1xxx_xxxxb</td> <td>Divide by 256.</td> </tr> <tr> <td>0000_1xxxb</td> <td>Divide by 16.</td> <td></td> <td></td> </tr> </tbody> </table> Host Controller Version 3.00: <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>3FFh-001h</td> <td>Divide by 1/(2*SdclkDiv[9:0])</td> </tr> <tr> <td>000h</td> <td>Divide by 1</td> </tr> </tbody> </table>	Bits	Definition	Bits	Definition	0000_0000b	Divide by 1.	0001_xxxxb	Divide by 32	0000_0001b	Divide by 2.	001x_xxxxb	Divide by 64.	0000_001xb	Divide by 4.	01xx_xxxxb	Divide by 128.	0000_01xxb	Divide by 8.	1xxx_xxxxb	Divide by 256.	0000_1xxxb	Divide by 16.			Bits	Definition	3FFh-001h	Divide by 1/(2*SdclkDiv[9:0])	000h	Divide by 1
Bits	Definition	Bits	Definition																												
0000_0000b	Divide by 1.	0001_xxxxb	Divide by 32																												
0000_0001b	Divide by 2.	001x_xxxxb	Divide by 64.																												
0000_001xb	Divide by 4.	01xx_xxxxb	Divide by 128.																												
0000_01xxb	Divide by 8.	1xxx_xxxxb	Divide by 256.																												
0000_1xxxb	Divide by 16.																														
Bits	Definition																														
3FFh-001h	Divide by 1/(2*SdclkDiv[9:0])																														
000h	Divide by 1																														
7:6	<b>SdclkDiv[9:8].</b> Read-write. Reset: 0.																														
5	<b>ClkGenSel: Clock Generator Select.</b> Read-write. Reset: 0. 1=Programmable Clock Mode. 0=Divided Clock Mode.																														
4:3	Reserved.																														
2	<b>SdclkEn.</b> Read-write. Reset: 0. SD Clock Enable. SDCLK Frequency Select can be changed when this bit is 0. When card is removed, this bit is cleared to 0 automatically. 0=Disable. 1=Enable.																														
1	<b>SysclkStable.</b> Read-only. Reset: 0. Internal Clock Stable. 0=Unstable. 1=Stable.																														
0	<b>SysclkEn.</b> Read-write. Reset: 0. 1= Internal Clock Enable. 0=Disable.																														

**SDHCx30 SDHC\_INT\_STATUS**

Bits	Description
31:30	Reserved.
29	<b>CeAtaErr.</b> Read; Write-1-to-clear. Reset: 0. CE-ATA Error.
28	<b>SdmaErr.</b> Read; Write-1-to-clear. Reset: 0. SDMA Error.
27	Reserved.
26	<b>TuningError.</b> Read; Write-1-to-clear. Reset: 0. 1=An unrecoverable tuning error is detected in a during the tuning procedure.
25	<b>AdmaErr.</b> Read; Write-1-to-clear. Reset: 0. ADMA Error.
24	<b>Acmd12Err.</b> Read; Write-1-to-clear. Reset: 0. Auto CMD12 Error. Logical OR of Auto CMD12 Error Status Register.
23	Reserved.
22	<b>DatEndErr.</b> Read; Write-1-to-clear. Reset: 0. Data CRC Error.
21	<b>DatCrcErr.</b> Read; Write-1-to-clear. Reset: 0. Data CRC Error.
20	<b>DatToErr.</b> Read; Write-1-to-clear. Reset: 0. Data Timeout Error.
19	<b>CmdIdxErr.</b> Read; Write-1-to-clear. Reset: 0. Command Index Error. Mismatch of Command Index and index of response.
18	<b>CmdEndErr.</b> Read; Write-1-to-clear. Reset: 0. Command End Bit Error.
17	<b>CmdCrcErr.</b> Read; Write-1-to-clear. Reset: 0. Command CRC Error. If both CMD_TO_ERR and CMD_CRC_ERR are set, this indicates Command Conflict Error.
16	<b>CmdToErr.</b> Read; Write-1-to-clear. Reset: 0. Command Timeout Error. Response not returned within 128 SDCLK cycles.
15	<b>Error.</b> Read; Write-1-to-clear. Reset: 0. Error Interrupt.
14:13	Reserved.
12	<b>ReTuningEvent.</b> Read-only. Reset: 0. 1=Re-tuning should be performed.
11:9	Reserved.
8	<b>Sdio.</b> Read; Write-1-to-clear. Reset: 0. SDIO Card Interrupt. Writing 1 to this register does not clear this bit. To clear this bit, interrupt factor of SDIO cards should be cleared. The value of this bit is latched internally as long as the Card Interrupt bit (D08) in the Normal Interrupt Status Enable register is 1.
7	<b>CardRem.</b> Read; Write-1-to-clear. Reset: 0. Card Removal.
6	<b>CardIns.</b> Read; Write-1-to-clear. Reset: 0. Card Insertion.
5	<b>BufRdRdy.</b> Read; Write-1-to-clear. Reset: 0. Buffer Read Ready. In the case where Auto CMD12 is enabled and last block has been transferred, Auto CMD12 will be issued prior to this bit being set to 1. Clearing this bit should be done before buffer reading, because SD controller has dual buffer and the next Buffer Read Ready interrupt may occur immediately.
4	<b>BufWrRdy.</b> Read; Write-1-to-clear. Reset: 0. Buffer Write Ready. Clearing this bit should be done before buffer writing, because the SD controller has dual buffer and the next Buffer Write Ready interrupt may occur immediately.



3	<b>DmaEvt.</b> Read; Write-1-to-clear. Reset: 0. DMA Interrupt. It is set when internal counter reaches the value designated by Host DMA Buffer Boundary. It should be cleared by Host Driver after System Address Register is updated.
2	<b>BlockGapEvt.</b> Read; Write-1-to-clear. Reset: 0. Block Gap Event. It indicates the timing of next block gap, which was requested by the Stop At Block Gap Request. In case of write transaction, this interrupt will be generated before busy completion.
1	<b>DatDone.</b> Read; Write-1-to-clear. Reset: 0. Data Transfer Complete. Indicates the timing for completion of data transaction, which includes the completion at the block gap by the Stop At Block Gap Request. When some errors are detected during data transaction, this bit will not be set. In the case where Auto CMD12 is enabled, Auto CMD12 will be issued prior to this bit being set to 1.
0	<b>CmdDone: cmdDone (r).</b> Read-write. Reset: 0. Command Complete. The end bit of the command response is received. In the case of commands with no response, the end of the command.

**SDHCx34 SDHC\_INT\_MASK**

Bits	Description
31:30	Reserved.
29	<b>CeAtaErrMask.</b> Read-write. Reset: 0. CE-ATA Error. 0=Masked. 1=Enable.
28	<b>SdmaErrMask.</b> Read-write. Reset: 0. SDMA Error. 0=Masked. 1=Enable.
27	Reserved.
26	<b>TuningErrorStatusEnable.</b> Read-write. Reset: 0. 1=Tuning Event Status Enable. 0=Masked.
25	<b>AdmaErrMask.</b> Read-write. Reset: 0. ADMA Error. 0=Masked. 1=Enable.
24	<b>Acmd12ErrMask.</b> Read-write. Reset: 0. Auto CMD12 Error. 0=Masked. 1=Enable.
23	<b>CurLimErrMask.</b> Read-write. Reset: 0. Current Limit Error. 0=Masked. 1=Enable.
22	<b>DatEndErrMask.</b> Read-write. Reset: 0. Data End Bit Error. 0=Masked. 1=Enable.
21	<b>DatCrcErrMask.</b> Read-write. Reset: 0. Data CRC Error. 0=Masked. 1=Enable.
20	<b>DatToErrMask.</b> Read-write. Reset: 0. Data Timeout Error. 0=Masked. 1=Enable.
19	<b>CmdIdxErrMask.</b> Read-write. Reset: 0. Command Index Error. 0=Masked. 1=Enable.
18	<b>CmdEndErrMask.</b> Read-write. Reset: 0. Command End Bit Error. 0=Masked. 1=Enable.
17	<b>CmdCrcErrMask.</b> Read-write. Reset: 0. Command CRC Error. 0=Masked. 1=Enable.
16	<b>CmdToErrMask.</b> Read-write. Reset: 0. Command Timeout Error. 0=Masked. 1=Enable.
15:13	Reserved.
12	<b>ReTuningEventStatusEn.</b> Read-write. Reset: 0. 1=Re-Tuning Event Status Enable. 0=Masked.
11:9	Reserved.
8	<b>SdioMask.</b> Read-write. Reset: 0. Card Interrupt. 0=Masked. 1=Enable.
7	<b>CardRemMask.</b> Read-write. Reset: 0. Card Removal. 0=Masked. 1=Enable.
6	<b>CardInsMask.</b> Read-write. Reset: 0. Card Insertion. 0=Masked. 1=Enable.
5	<b>BufRdRdyMask.</b> Read-write. Reset: 0. Buffer Read Ready. 0=Masked. 1=Enable.
4	<b>BufWrRdyMask.</b> Read-write. Reset: 0. Buffer Write Ready. 0=Masked. 1=Enable.
3	<b>DmaEvtMask.</b> Read-write. Reset: 0. DMA Interrupt. 0=Masked. 1=Enable.
2	<b>BlockGapEvtMask.</b> Read-write. Reset: 0. Block Gap Event. 0=Masked. 1=Enable.

1	<b>DatDoneMask.</b> Read-write. Reset: 0. Transfer Complete. 0=Masked. 1=Enable.
0	<b>CmdDoneMask.</b> Read-write. Reset: 0. Command Complete. 0=Masked. 1=Enable.

**SDHCx38 SDHC\_SIG\_MASK**

Bits	Description
31:30	Reserved.
29	<b>CeAtaErrEn.</b> Read-write. Reset: 0. CE-ATA Error. 0=Masked. 1=Enable.
28	<b>SdmaErrEn.</b> Read-write. Reset: 0. SDMA Error. 0=Masked. 1=Enable.
27	Reserved.
26	<b>TuningErrorStatusEnable.</b> Read-write. Reset: 0. 1=Tuning Event Status Enable. 0=Masked.
25	<b>AdmaErrEn.</b> Read-write. Reset: 0. ADMA Error. 0=Masked. 1=Enable.
24	<b>Acmd12ErrEn.</b> Read-write. Reset: 0. Auto CMD12 Error. 0=Masked. 1=Enable.
23	<b>CurLimErrEn.</b> Read-write. Reset: 0. Current Limit Error. 0=Masked. 1=Enable.
22	<b>DatEndErrEn.</b> Read-write. Reset: 0. Data End Bit Error. 0=Masked. 1=Enable.
21	<b>DatCrcErrEn.</b> Read-write. Reset: 0. Data CRC Error. 0=Masked. 1=Enable.
20	<b>DatToErrEn.</b> Read-write. Reset: 0. Data Timeout Error. 0=Masked. 1=Enable.
19	<b>CmdIdxErrEn.</b> Read-write. Reset: 0. Command Index Error. 0=Masked. 1=Enable.
18	<b>CmdEndErrEn.</b> Read-write. Reset: 0. Command End Bit Error. 0=Masked. 1=Enable.
17	<b>CmdCrcErrEn.</b> Read-write. Reset: 0. Command CRC Error. 0=Masked. 1=Enable.
16	<b>CmdToErrEn.</b> Read-write. Reset: 0. Command Timeout Error. 0=Masked. 1=Enable.
15:13	Reserved.
12	<b>ReTuningEventStatusEn.</b> Read-write. Reset: 0. 1=Re-Tuning Event Status Enable. 0=Masked.
11:9	Reserved.
8	<b>SdioEn.</b> Read-write. Reset: 0. Card Interrupt. 0=Masked. 1=Enable.
7	<b>CardRemEn.</b> Read-write. Reset: 0. Card Removal. 0=Masked. 1=Enable.
6	<b>CardInsEn.</b> Read-write. Reset: 0. Card Insertion. 0=Masked. 1=Enable.
5	<b>BufRdRdyEn.</b> Read-write. Reset: 0. Buffer Read Ready. 0=Masked. 1=Enable.
4	<b>BufWrRdyEn.</b> Read-write. Reset: 0. Buffer Write Ready. 0=Masked. 1=Enable.
3	<b>DmaEvtEn.</b> Read-write. Reset: 0. DMA Interrupt. 0=Masked. 1=Enable.
2	<b>BlockGapEvtEn.</b> Read-write. Reset: 0. Block Gap Event. 0=Masked. 1=Enable.
1	<b>DatDoneEn.</b> Read-write. Reset: 0. Transfer Complete. 0=Masked. 1=Enable.
0	<b>CmdDoneEn.</b> Read-write. Reset: 0. Command Complete. 0=Masked. 1=Enable.

**SDHCx3C SDHC ACMD12 Error/Host Control 2**

Bits	Description
31	<b>PresetValEn.</b> Read-write. Reset: 0. 1=Enable automatic selection of Preset Value. 0=SDCLK and driver strength are controlled by Host Driver.

30	<b>AsyncIntEn.</b> Read-write. Reset: 0. Enable Asynchronous Interrupts.																
29:24	Reserved.																
23	<b>SamplingClkSel.</b> Read-write. Reset: 0. 1=Tuned clock is used to sample data. 0=Fixed clock is used to sample data.																
22	<b>ExecTuning.</b> Read-write; cleared-when-done. Reset: 0. 1=Execute Tuning. 0=Not Tuned or Tuning Completed.																
21:20	<b>DvrStrengthSel.</b> Read-write. Reset: 0. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Driver Type B is Selected (Default)</td> <td>10b</td> <td>Driver Type C is Selected</td> </tr> <tr> <td>01b</td> <td>Driver Type A is Selected</td> <td>11b</td> <td>Driver Type D is Selected</td> </tr> </tbody> </table>	Bits	Definition	Bits	Definition	00b	Driver Type B is Selected (Default)	10b	Driver Type C is Selected	01b	Driver Type A is Selected	11b	Driver Type D is Selected				
Bits	Definition	Bits	Definition														
00b	Driver Type B is Selected (Default)	10b	Driver Type C is Selected														
01b	Driver Type A is Selected	11b	Driver Type D is Selected														
19	<b>En_1_8VSignaling.</b> Read-write. Reset: 0. 1=1.8 Volt Signaling. 0=3.3 Volt Signaling.																
18:16	<b>UHS_ModeSel.</b> Read-write. Reset: 0. Select UHS-I mode when (En_1_8VSignaling==1). <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>SDR12</td> <td>011b</td> <td>SDR104</td> </tr> <tr> <td>001b</td> <td>SDR25</td> <td>100b</td> <td>DDR50</td> </tr> <tr> <td>010b</td> <td>SDR50</td> <td>111b-101b</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Definition	Bits	Definition	000b	SDR12	011b	SDR104	001b	SDR25	100b	DDR50	010b	SDR50	111b-101b	Reserved
Bits	Definition	Bits	Definition														
000b	SDR12	011b	SDR104														
001b	SDR25	100b	DDR50														
010b	SDR50	111b-101b	Reserved														
15:8	Reserved.																
7	<b>CmdErr.</b> Read-only. Reset: 0. Command Not Issued By Auto CMD12 Error.																
6:5	Reserved.																
4	<b>IndexErr.</b> Read-only. Reset: 0. Auto CMD12 Index Error.																
3	<b>EndErr.</b> Read-only. Reset: 0. Auto CMD12 End Bit Error.																
2	<b>CrcErr.</b> Read-only. Reset: 0. Auto CMD12 CRC Error.																
1	<b>ToErr.</b> Read-only. Reset: 0. Auto CMD12 Timeout Error.																
0	<b>ExeErr.</b> Read-only. Reset: 0. Auto CMD12 Not Executed Error.																

**SDHCx40 SDHC\_CAPABILITY**

Bits	Description												
31:30	<b>SlotType.</b> Read-only. Reset: 0. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Removable Card Slot</td> <td>10b</td> <td>Shared Bus Slot</td> </tr> <tr> <td>01b</td> <td>Embedded Slot for One Device</td> <td>11b</td> <td>Reserved.</td> </tr> </tbody> </table>	Bits	Definition	Bits	Definition	00b	Removable Card Slot	10b	Shared Bus Slot	01b	Embedded Slot for One Device	11b	Reserved.
Bits	Definition	Bits	Definition										
00b	Removable Card Slot	10b	Shared Bus Slot										
01b	Embedded Slot for One Device	11b	Reserved.										
29	<b>AsynIntSup: Asynchronous Interrupt Support.</b> Read-only. Reset: 1. 1=Asynchronous interrupt supported. 0=Asynchronous interrupt not supported.												
28	<b>Support64BitSystemBus.</b> Read-only. Reset: 0. 1=Host Controller supports 64-bit address descriptor mode and 64-bit address system bus.												
27	Reserved.												
26	<b>Support18v.</b> Read-only. Reset: 1. 1=Voltage Support for 1.8V.												
25	<b>Support30v.</b> Read-only. Reset: 0. 1=Voltage Support for 3.0V.												
24	<b>Support33v.</b> Read-only. Reset: 1. 1=Voltage Support for 3.3V.												
23	<b>SusResSupport.</b> Read-only. Reset: 1. 1=Suspend and Resume is supported.												
22	<b>DmaSupport.</b> Read-only. Reset: 1. 1=DMA is supported.												

21	<b>HiSpeedSupport.</b> Read-only. Reset: 1. 1=High Speed is supported.												
20	<b>AdmaSupport.</b> Read-only. Reset: 1. 1=Advanced DMA is supported.												
19	<b>Adma2Support.</b> Read-only. Reset: 1. 1=Advanced DMA2 is supported.												
18	<b>Mmc8Support.</b> Read-only. Reset: 0. Extended Media Bus Support (MMC).												
17:16	<p><b>MaxBlkLen.</b> Read-only. Reset: 2h. Specifies the maximum block size that the Host Driver can read and write to the buffer in the Host Controller. The buffer shall transfer this block size without wait cycles.</p> <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>512 bytes</td> <td>10b</td> <td>2048 bytes</td> </tr> <tr> <td>01b</td> <td>1024 bytes</td> <td>11b</td> <td>Reserved</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	<u>Bits</u>	<u>Definition</u>	00b	512 bytes	10b	2048 bytes	01b	1024 bytes	11b	Reserved
<u>Bits</u>	<u>Definition</u>	<u>Bits</u>	<u>Definition</u>										
00b	512 bytes	10b	2048 bytes										
01b	1024 bytes	11b	Reserved										
15:8	<p><b>BaseClkFreq.</b> Read-only. Reset: 32h. Specifies the base clock frequency for SD Clock.</p> <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>00h</td> <td>Reserved</td> </tr> <tr> <td>FFh-01h</td> <td>&lt;BaseClkFreq&gt; MHz</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	00h	Reserved	FFh-01h	<BaseClkFreq> MHz						
<u>Bits</u>	<u>Definition</u>												
00h	Reserved												
FFh-01h	<BaseClkFreq> MHz												
7	<b>TmoClkUnit: Timeout Clock Unit.</b> Read-only. Reset: 1. Specifies the unit of TmoClkFreq. 0=Unit is KHz. 1=Unit is MHz.												
6	Reserved.												
5:0	<b>TmoClkFreq: Timeout Clock Frequency.</b> Read-only. Reset: 32h. Specifies the base clock frequency used to detect Data Timeout Error. TmoClkUnit defines the unit of this field's value.												

**SDHCx44 SDHC\_CAPABILITY 2**

Bits	Description										
31:24	Reserved.										
23:16	<p><b>ClockMultiplier.</b> Read-only. Reset: 3.</p> <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>00h</td> <td>ClockMultiplier not supported.</td> </tr> <tr> <td>FFh-01h</td> <td>&lt;ClockMultiplier+1&gt; MHz</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	00h	ClockMultiplier not supported.	FFh-01h	<ClockMultiplier+1> MHz				
<u>Bits</u>	<u>Definition</u>										
00h	ClockMultiplier not supported.										
FFh-01h	<ClockMultiplier+1> MHz										
15:14	<p><b>ReTuningModes.</b> Read-only. Reset: 0. Specifies re-tuning method and limits the maximum data length.</p> <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Mode 1, Timer, 4 MBMax Data length.</td> </tr> <tr> <td>01b</td> <td>Mode 2, Timer and Re-Tuning Request, 4 MBMax Data length.</td> </tr> <tr> <td>10b</td> <td>Mode 3, Auto Re-Tuning (for transfer)Timer and Re-Tuning Request, any Data Length.</td> </tr> <tr> <td>11b</td> <td>Reserved.</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	00b	Mode 1, Timer, 4 MBMax Data length.	01b	Mode 2, Timer and Re-Tuning Request, 4 MBMax Data length.	10b	Mode 3, Auto Re-Tuning (for transfer)Timer and Re-Tuning Request, any Data Length.	11b	Reserved.
<u>Bits</u>	<u>Definition</u>										
00b	Mode 1, Timer, 4 MBMax Data length.										
01b	Mode 2, Timer and Re-Tuning Request, 4 MBMax Data length.										
10b	Mode 3, Auto Re-Tuning (for transfer)Timer and Re-Tuning Request, any Data Length.										
11b	Reserved.										
13	<b>UseTuningForSDR50.</b> Read-only. Reset: 0. 1=SDR50 requires tuning.										
12	Reserved.										

11:8	<b>TimerCountForReTuning.</b> Read-only. Reset: 5. Specifies value of the Re-Tuning Timer for Re-Tuning Mode 1 to 3. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Re-tuning timer disabled.</td> </tr> <tr> <td>1h</td> <td>1 Second</td> </tr> <tr> <td>Bh-2h</td> <td>&lt;2^(TimerCountForReTuning-1)&gt; Seconds</td> </tr> <tr> <td>Eh-Ch</td> <td>Reserved</td> </tr> <tr> <td>Fh</td> <td>Get information from other source.</td> </tr> </tbody> </table>	Bits	Definition	0h	Re-tuning timer disabled.	1h	1 Second	Bh-2h	<2^(TimerCountForReTuning-1)> Seconds	Eh-Ch	Reserved	Fh	Get information from other source.
Bits	Definition												
0h	Re-tuning timer disabled.												
1h	1 Second												
Bh-2h	<2^(TimerCountForReTuning-1)> Seconds												
Eh-Ch	Reserved												
Fh	Get information from other source.												
7	Reserved.												
6	<b>DriverTypeDSupport.</b> Read-only. Reset: 0. 1=Support of Driver Type D for 1.8 Signaling.												
5	<b>DriverTypeCSupport.</b> Read-only. Reset: 0. 1=Support of Driver Type C for 1.8 Signaling.												
4	<b>DriverTypeASupport.</b> Read-only. Reset: 0. 1=Support of Driver Type A for 1.8 Signaling.												
3	Reserved.												
2	<b>DDR50Support.</b> Read-only. Reset: 0. 1=DDR50 is supported.												
1:0	<b>SDRSupport.</b> Read-only. Reset: 3. <table border="1"> <thead> <tr> <th>Bit</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>[0]</td> <td>1=SDR50 is supported.</td> </tr> <tr> <td>[1]</td> <td>1=SDR104 is supported.</td> </tr> </tbody> </table>	Bit	Definition	[0]	1=SDR50 is supported.	[1]	1=SDR104 is supported.						
Bit	Definition												
[0]	1=SDR50 is supported.												
[1]	1=SDR104 is supported.												

**SDHCx48 SDHC\_CURR\_CAPABILITY**

Bits	Description
31:24	Reserved.
23:16	<b>MaxCurr18v.</b> Read-only. Reset: C8h. Max Current for 1.8V.
15:8	<b>MaxCurr30v.</b> Read-only. Reset: 0. Max Current for 3.0V.
7:0	<b>MaxCurr33v.</b> Read-only. Reset: 64h. Max Current for 3.3V.

**SDHCx50 SDHC\_FORCE\_EVT**

Bits	Description
31:30	Reserved.
29	<b>CeAtaErrFrc.</b> Read-write. Reset: 0. Force CE-ATA Error.
28	<b>SdmaErrFrc.</b> Read-write. Reset: 0. Force SDMA Error.
27:26	Reserved.
25	<b>AdmaErrFrc.</b> Read-write. Reset: 0. Force ADMA Error.
24	<b>Acmd12ErrFrc.</b> Read-write. Reset: 0. Force Auto CMD12 Error.
23	<b>CurLimErrFrc.</b> Read-write. Reset: 0. Force Current Limit Error.
22	<b>DatEndErrFrc.</b> Read-write. Reset: 0. Force Data End Bit Error.
21	<b>DatCrcErrFrc.</b> Read-write. Reset: 0. Force Data CRC Error.
20	<b>DatToErrFrc.</b> Read-write. Reset: 0. Force Data Timeout Error.
19	<b>CmdIdxErrFrc.</b> Read-write. Reset: 0. Force Command Index Error.

18	<b>CmdEndErrFrc.</b> Read-write. Reset: 0. Force Command End Bit Error.
17	<b>CmdCrcErrFrc.</b> Read-write. Reset: 0. Force Command CRC Error.
16	<b>CmdToErrFrc.</b> Read-write. Reset: 0. Force Command Timeout Error.
15:8	Reserved.
7	<b>Acmd12CmdErrFrc.</b> Read-write. Reset: 0. Force Command Not Issued By Auto CMD12 Error.
6:5	Reserved.
4	<b>Acmd12IdxErrFrc.</b> Read-write. Reset: 0. Force Auto CMD12 Index Error.
3	<b>Acmd12EndErrFrc.</b> Read-write. Reset: 0. Force Auto CMD12 End Bit Error.
2	<b>Acmd12CrcErrFrc.</b> Read-write. Reset: 0. Force Auto CMD12 CRC Error.
1	<b>Acmd12ToErrFrc.</b> Read-write. Reset: 0. Force Auto CMD12 Timeout Error.
0	<b>Acmd12ExeErrFrc.</b> Read-write. Reset: 0. Force Auto CMD12 Not Executed Error.

**SDHCx54 SDHC\_ADMA\_ERR**

Bits	Description										
31:3	Reserved.										
2	<b>AddrLenMismatch.</b> Read-only. Reset: 0. ADMA Address Length Mismatch Error. This error occurs in the following 2 cases:1. While Block Count Enable being set, the total data length specified by the descriptor table is different from that specified by the Block Count and Block Length.2. Total data length cannot be divided by the block length.										
1:0	<b>AdmaState.</b> Read-only. Reset: 0. ADMA State when error has occurred. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Stop DMA</td> </tr> <tr> <td>01b</td> <td>Fetch Descriptor.</td> </tr> <tr> <td>10b</td> <td>Change Address.</td> </tr> <tr> <td>11b</td> <td>Transfer Data.</td> </tr> </tbody> </table>	Bits	Definition	00b	Stop DMA	01b	Fetch Descriptor.	10b	Change Address.	11b	Transfer Data.
Bits	Definition										
00b	Stop DMA										
01b	Fetch Descriptor.										
10b	Change Address.										
11b	Transfer Data.										

**SDHCx58 SDHC\_ADMA\_SAD**

Bits	Description
31:16	<b>AdmaSysAddr1.</b> Read-write. Reset: 0. Upper bits. ADMA System Address. Before ADMA data transfer, the descriptor address should be set by the Host Driver. This address needs to be set with 4-byte alignment, since the descriptor table has 32-bit (4 byte) information formatted.
15:0	<b>AdmaSysAddr0.</b> Read-write. Reset: 0. Lower bits.

**SDHCx6[C:0:step4] Preset Value**

Table 235: Register Mapping for SDHCx6[C:0:step4]

Register	Function
SDHCx60	Initialization 3.3V or 1.8V/Default Speed 3.3V

Table 235: Register Mapping for SDHCx6[C:0:step4]

SDHCx64	High Speed 3.3V/SDR12 1.8V Preset Values
SDHCx68	SDR25 1.8V/SDR50 1.8V Preset Values
SDHCx6C	SDR104 1.8V/DDR50 1.8V Preset Values

Table 236: Field Mapping for SDHCx6[C:0:step4]

Register	Bits					
	31:30	26	25:16	15:14	10	9:0
SDHCx60	DefaultSpeed	DefaultSpeed	DefaultSpeed	Initialization	Initialization	Initialization
SDHCx64	SDR12	SDR12	SDR12	HighSpeed	HighSpeed	HighSpeed
SDHCx68	SDR50	SDR50	SDR50	SDR25	SDR25	SDR25
SDHCx6C	DDR50	DDR50	DDR50	SDR104	SDR104	SDR104

Bits	Description												
31:30	<b>DvrStrengthSelPreset1.</b> See: DvrStrengthSelPreset0.												
29:27	Reserved.												
26	<b>ClkGenSelPreset1.</b> See: ClkGenSelPreset0.												
25:16	<b>SdClkFreqSel1.</b> See: SdClkFreqSel0.												
15:14	<b>DvrStrengthSelPreset0.</b> Read-only. Driver Strength is supported by 1.8V signaling bus speed modes. Does not apply for 3.3V signaling. See <a href="#">SDHCx3C[DvrStrengthSel]</a> . <table border="0" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;"><u>Bits</u></td> <td style="text-align: center;"><u>Definition</u></td> <td style="text-align: center;"><u>Bits</u></td> <td style="text-align: center;"><u>Definition</u></td> </tr> <tr> <td style="text-align: center;">00b</td> <td>Driver Type B is Selected</td> <td style="text-align: center;">10b</td> <td>Driver Type C is Selected</td> </tr> <tr> <td style="text-align: center;">01b</td> <td>Driver Type A is Selected</td> <td style="text-align: center;">11b</td> <td>Driver Type D is Selected</td> </tr> </table>	<u>Bits</u>	<u>Definition</u>	<u>Bits</u>	<u>Definition</u>	00b	Driver Type B is Selected	10b	Driver Type C is Selected	01b	Driver Type A is Selected	11b	Driver Type D is Selected
<u>Bits</u>	<u>Definition</u>	<u>Bits</u>	<u>Definition</u>										
00b	Driver Type B is Selected	10b	Driver Type C is Selected										
01b	Driver Type A is Selected	11b	Driver Type D is Selected										
13:11	Reserved.												
10	<b>ClkGenSelPreset0.</b> Read-only. Clock Generator Select Value. See <a href="#">SDHCx2C[ClkGenSel]</a> . 1=Programmable Clock Generator. 0=Host Controller Ver2.00 Compatible Clock Generator.												
9:0	<b>SdClkFreqSel0.</b> Read-only. A10-bit preset value. See <a href="#">SDHCx2C[SdclkDiv[9:0]]</a> .												

**SDHCxE0 Shared Bus Control Register**

Bits	Description																
31	Reserved.																
30:24	<b>BackEndPwrCtrl: Back-End Power Control.</b> Read-write. Reset: 0. 0=Back-end Power is Off. 1=Back-end Power is Supplied. <table border="0" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;"><u>Bit</u></td> <td style="text-align: center;"><u>Definition</u></td> </tr> <tr> <td style="text-align: center;">[0]</td> <td>Back-end Power Control for Device 1</td> </tr> <tr> <td style="text-align: center;">[1]</td> <td>Back-end Power Control for Device 2</td> </tr> <tr> <td style="text-align: center;">[2]</td> <td>Back-end Power Control for Device 3</td> </tr> <tr> <td style="text-align: center;">[3]</td> <td>Back-end Power Control for Device 4</td> </tr> <tr> <td style="text-align: center;">[4]</td> <td>Back-end Power Control for Device 5</td> </tr> <tr> <td style="text-align: center;">[5]</td> <td>Back-end Power Control for Device 6</td> </tr> <tr> <td style="text-align: center;">[6]</td> <td>Back-end Power Control for Device 7</td> </tr> </table>	<u>Bit</u>	<u>Definition</u>	[0]	Back-end Power Control for Device 1	[1]	Back-end Power Control for Device 2	[2]	Back-end Power Control for Device 3	[3]	Back-end Power Control for Device 4	[4]	Back-end Power Control for Device 5	[5]	Back-end Power Control for Device 6	[6]	Back-end Power Control for Device 7
<u>Bit</u>	<u>Definition</u>																
[0]	Back-end Power Control for Device 1																
[1]	Back-end Power Control for Device 2																
[2]	Back-end Power Control for Device 3																
[3]	Back-end Power Control for Device 4																
[4]	Back-end Power Control for Device 5																
[5]	Back-end Power Control for Device 6																
[6]	Back-end Power Control for Device 7																
23	Reserved.																

22:20	<b>IntPinSel.</b> Read-write. Reset: 0. Selects interrupt pin inputs.			
	<u>Bits</u>	<u>Definition</u>	<u>Bits</u>	<u>Definition</u>
	000b	Interrupt is detected by Interrupt Cycle	x1xb	INT_B is Enabled
	xx1b	INT_A is Enabled	1xxb	INT_C is Enabled
19	Reserved.			
18:16	<b>ClkPinSel.</b> Read-write. Reset: 0. Selects one of clock pin outputs.			
	<u>Bits</u>	<u>Definition</u>	<u>Bits</u>	<u>Definition</u>
	000b	Clock Pins are Disabled	100b	CLK[4] is Selected
	001b	CLK[1] is Selected	101b	CLK[5] is Selected
	010b	CLK[2] is Selected	110b	CLK[6] is Selected
	011b	CLK[3] is Selected	111b	CLK[7] is Selected
15	Reserved.			
14:8	<b>BusWidthPreset.</b> Read-only. 0=4-bit bus width mode (Data Transfer Width = 1). 1=8-bit bus width mode.			
	<u>Bit</u>	<u>Definition</u>		
	[0]	Bus Width Preset for Device 1		
	[1]	Bus Width Preset for Device 2		
	[2]	Bus Width Preset for Device 3		
	[3]	Bus Width Preset for Device 4		
	[4]	Bus Width Preset for Device 5		
	[5]	Bus Width Preset for Device 6		
	[6]	Bus Width Preset for Device 7		
7:6	Reserved.			
5:4	<b>NumIntInputPins.</b> Read-only. Indicates support of interrupt input pins INT_A#, INT_B# and INT_C# for shared bus system.			
	<u>Bits</u>	<u>Definition</u>	<u>Bits</u>	<u>Definition</u>
	00b	Interrupt Input Pin is Not Supported	10b	INTA and INTB are Supported
	01b	INTA is Supported	11b	INTA, INTB and INTC are Supported
3	Reserved.			
2:0	<b>NumofClkPins.</b> Read-only. This field indicates support of clock pins to select one of devices for shared bus system.			
	<u>Bits</u>	<u>Definition</u>	<u>Bits</u>	<u>Definition</u>
	000b	Shared bus is not supported	100b	4 SDCLK pin is supported
	001b	1 SDCLK pin is supported	101b	5 SDCLK pin is supported
	010b	2 SDCLK pin is supported	110b	6 SDCLK pin is supported
	011b	3 SDCLK pin is supported	111b	7 SDCLK pin is supported

**SDHCxFC SDHC\_VER\_SLOT**

Bits	Description
31:24	<b>VendorVersion.</b> Read-only. Reset: 0. Vendor version.
23:16	<b>SpecVersion.</b> Read-only. Reset: 0. Specification version.



15:8	Reserved.
7:0	<b>SlotIntrpt.</b> Read-only. Reset: 0. Interrupt Signal for Each Slot. The value of XSLT_INT7-0 inputs, which indicates the logical OR of Interrupt signal and Wakeup signal, are inverted and referred to by this register. In case of multiple slots, Interrupt signal and Wakeup signal should be logical OR'ed externally and should be inputted to each of XSLT_INT7-0.

### 3.25.7 SMBus Host Controller

#### 3.25.7.1 Device 14h Function 0 (SMBus) 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.

#### D14F0x00 Device/Vendor ID

Bits	Description
31:16	<b>DeviceID: Device Identifier.</b> Value: 780Bh. This 16-bit field is assigned by the device manufacturer and identifies the type of device.
15:0	<b>VendorID.</b> Read-only. Reset: 1022h. Vendor Identifier.

#### D14F0x04 Status/Command

Bits	Description
31	<b>DetectedParityError: Detected Parity Error.</b> Read; write-1-to-clear. Reset: 0. This bit is set by the device whenever it detects a parity error, even if parity error handling is disabled.
30	<b>SignaledSystemError: Signaled System Error.</b> Read; write-1-to-clear. Reset: 0. This bit is set by the device whenever the device asserts SERR#.
29	<b>ReceivedMasterAbort: Received Master Abort.</b> Read; write-1-to-clear. Reset: 0. This bit is set by a slave device whenever it terminates its transaction with master abort.
28	<b>ReceivedTargetAbort: Received Target Abort.</b> Read; write-1-to-clear. Reset: 0. This bit is set by a master device whenever its transaction is terminated with target abort.
27	<b>SignaledTargetAbort: Signaled Target Abort.</b> Read; write-1-to-clear. Reset: 0. This bit is set by a slave device whenever it terminates a cycle with target abort.
26:25	<b>DEVSELTiming: DEVSEL Timing.</b> Value: 1. These bits encode the timing of DEVSEL#. This module always responds in medium timing.
24	<b>DataParityErrorDetected: Data Parity Error Detected.</b> Read; write-1-to-clear. Reset: 0. 1=The Parity Error Response bit is set and the module has detected PERR# asserted while acting as a PCI master regardless of whether PERR# was driven by this module.
23	<b>FastBack2BackCapable: Fast Back to Back Capable.</b> Value: 0. 0=Fast back to back cycles are not supported.
22	<b>UDFSupported: UDF Supported.</b> Value: 0. 0=User definable feature is not supported.
21	<b>Capable66MHz: 66 MHz Capable.</b> Value: 1. 1=This device is 66 MHz capable.
20	<b>MsiMappingCapability: MSI Mapping Capability.</b> Value: 1. 1=This device supports MSI mapping.
19:10	Reserved.
9	<b>FastBack2BackEnable: Fast Back-to-Back Enable.</b> Value: 0. ACPI/SMBus does not support fast back-to-back.
8	<b>SERREnable: System Error Enable.</b> Read-write. Reset: 0. 1=Enable system error reporting.
7	<b>WaitCycleControl: Wait Cycle Control.</b> Value: 0. 0=This module does not use address stepping.

6	<b>ParityErrorResponse: Parity Error Response.</b> Read-write. Reset: 0. This bit controls the device's response to parity errors. 1=The device must take its normal action when a parity error is detected. 0=The device must ignore any parity errors that it detects and continue normal operation.
5	<b>VGAPaletteSnoop: VGA Palette Snoop.</b> Value: 0. This bit does not apply to this module.
4	<b>MemoryWriteInvalidateEnable: Memory Write and Invalidate Enable.</b> Value: 0. This module does not generate Memory Write and Invalidate command.
3	<b>SpecialCycle: Special Cycle.</b> Value: 0. This module does not respond to special cycle.
2	<b>BusMaster: Bus Master.</b> Value: 0. ACPI/SMBus does not have PCI master.
1	<b>MemorySpace: Memory Space.</b> Read-write. Reset: 1. 1=Enable the device to respond to memory space accesses.
0	<b>IOSpace: IO Space.</b> Read-write. Reset: 1. 1=Enable the device to respond to IO space accesses.

**D14F0x08 Revision ID/Class Code**

Bits	Description
31:8	<b>ClassCode: Class Code.</b> Value: C0500h. Specifies a SMBUS controller.
7:0	<b>RevisionID: Revision ID.</b> Value: IF (RevA0) THEN 39h. ELSEIF (RevA1) THEN 3Ah. ENDIF.

**D14F0x0C Cache Line Size**

Bits	Description
31:24	<b>Bist.</b> Value: 0. The module has no built-in self-test and so this is always 0.
23:16	<b>HeaderType: Header Type.</b> Value: 80h. This device is a multifunction device.
15:8	<b>LatencyTimer: Latency Timer.</b> Value: 0. This register specifies the value of the Latency Timer. This is not used in this module and so it is always 0.
7:0	<b>CacheLineSize: Cache Line Size.</b> Value: 0. This register specifies the system cache line size. This module does not use Memory Write and Invalidate command and so this register is not applicable. It is hard coded to 0.

**D14F0x10 Base Address 0**

Bits	Description
31:0	<b>BaseAddress0: Base Address 0.</b> Value: 0. Not used and is hard coded to 0.

**D14F0x14 Base Address 1**

Bits	Description
31:0	<b>BaseAddress1: Base Address 1.</b> Value: 0. Not used and is hard coded to 0.

**D14F0x18 Base Address 2**

Bits	Description
31:0	<b>BaseAddress2: Base Address 2.</b> Value: 0. Not used and is hard coded to 0.

**D14F0x1C Base Address 3**

Bits	Description
31:0	<b>BaseAddress3: Base Address 3.</b> Value: 0. Not used and is hard coded to 0.

**D14F0x20 Base Address 4**

Bits	Description
31:0	<b>BaseAddress4: Base Address 4.</b> Value: 0. Not used and is hard coded to 0.

**D14F0x24 Base Address 5**

Bits	Description
31:0	<b>BaseAddress5: Base Address 5.</b> Value: 0. Not used and is hard coded to 0.

**D14F0x28 Cardbus CIS Pointer**

Bits	Description
31:0	<b>CardbusCISPointer: Cardbus CIS Pointer.</b> Value: 0. Not used and is hard coded to 0.

**D14F0x2C Subsystem Vendor ID**

Bits	Description
31:16	<b>SubsystemID: Subsystem ID.</b> Write-once. Reset: 780Bh.
15:0	<b>SubsystemVendorID: Subsystem Vendor ID.</b> Write-once. Reset: 1022h.

**D14F0x30 Expansion ROM Base Address**

Bits	Description
31:8	Reserved.
7:0	<b>ExpansionROMBaseAddress: Expansion ROM Base Address.</b> Value: 0. Not used.

**D14F0x34 Capability Pointer**

Bits	Description
31:8	Reserved.
7:0	<b>CapabilityPointer: Capability Pointer.</b> Read-only. Reset: 0.

**D14F0x3C Interrupt Line**

Bits	Description
31:24	<b>MaxLat.</b> Value: 0. Device has no requirements for the setting.
23:16	<b>MinGnt.</b> Value: 0. Device has no requirements for the setting.
15:8	<b>InterruptPin: Interrupt Pin.</b> Value: 0. 0=This module does not generate interrupts.
7:0	<b>InterruptLine: Interrupt Line.</b> Value: 0. This module does not generate interrupts.

**3.25.7.2 ASF (Alert Standard Format) Registers**

ASF register space is accessed through two methods:

- IO access through ASF IO base address defined by [PMx28 \[AsfEn\]](#):
  - A. Program the base address of ASF IO space through [PMx28\[AsfIoBase\]](#).
  - B. Enable ASF function and IO decoding through [PMx28\[AsfEn\]](#).
- Direct memory mapped or IO mapped access through the AcpiMmio region. The ASF registers range from [PMx24\[AcpiMmioAddr\]+900h](#) to [PMx24\[AcpiMmioAddr\]+9FFh](#). See [PMx24 \[AcpiMmioEn\]](#).

**ASFx00 HostStatus**

Bits	Description
7	<b>LastByte.</b> Read; set-by-hardware; write-1-to-clear. Reset: 0. 0=Last byte has not received. 1=Last byte has received.
6:5	Reserved.
4	<b>PECError.</b> Read; set-by-hardware; write-1-to-clear. Reset: 0. 0=No CRC error. 1=CRC error happened.
3	<b>BusCollision.</b> Read; set-by-hardware; write-1-to-clear. Reset: 0. 0=No bus collision. 1=Bus collision.
2	<b>DevError.</b> Read; set-by-hardware; write-1-to-clear. Reset: 0. 0=Slave device behaves correctly. 1=No ACK or slave device responses incorrectly.
1	<b>Intr.</b> Read; set-by-hardware; write-1-to-clear. Reset: 0. 1=Termination of a command. Whenever [HostBusy] is cleared, this bit is set.
0	<b>HostBusy.</b> Read-only; updated-by-hardware. Reset: 0. This bit reports the status of the ASF host. 0=SM bus host is idle. 1=SM bus host is busy.

**ASFx02 HostControl**

Bits	Description																		
7	<b>PECEnable.</b> Read-write. Reset: 0. 0=PEC disable. 1=PEC enable. Enable CRC checking when ASF HC presents as SM master and SM slave.																		
6	<b>Start.</b> Write-1-only; Read. Reset: 0. 1=Writing 1 to initiate the command. Always read back as 0.																		
5	<b>PECAppend.</b> Read-write. Reset: 0. 0=No PEC append. 1=Automatic PEC append. ASF HC calculates CRC code and append to the tail of the data packets.																		
4:2	<b>Protocol.</b> Read-write. Reset: 0. <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>Quick</td> </tr> <tr> <td>001b</td> <td>Byte</td> </tr> <tr> <td>010b</td> <td>Byte data</td> </tr> <tr> <td>011b</td> <td>Word data</td> </tr> <tr> <td>100b</td> <td>Process call</td> </tr> <tr> <td>101b</td> <td>Block</td> </tr> <tr> <td>110b</td> <td>Block write-block read-process call</td> </tr> <tr> <td>111b</td> <td>Reserved</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	000b	Quick	001b	Byte	010b	Byte data	011b	Word data	100b	Process call	101b	Block	110b	Block write-block read-process call	111b	Reserved
<u>Bits</u>	<u>Definition</u>																		
000b	Quick																		
001b	Byte																		
010b	Byte data																		
011b	Word data																		
100b	Process call																		
101b	Block																		
110b	Block write-block read-process call																		
111b	Reserved																		
1	<b>KillHost.</b> Read-write. Reset: 0. 0=Enable SM master. 1=Reset SM master.																		
0	Reserved.																		

**ASFx03 HostCommand**

Bits	Description
7:0	<b>HostCommand.</b> Read-write. Reset: 0. Command to be transmitted by master.

**ASFx04 SlaveAddress**

Bits	Description
7:1	<b>Address.</b> Read-write. Reset: 0. Provide the SM address of slave.
0	<b>RW.</b> Read-write. Reset: 0. 0=Write. 1=Read.

**ASFx05 Data0**

Bits	Description
7:0	<b>Data0.</b> Read-write. Reset: 0. Contains count or DATA0 field of transaction.

**ASFx06 Data1**

Bits	Description
7:0	<b>Data1.</b> Read-write. Reset: 0. Contains DATA1 field of transaction.

**ASFx07 DataIndex**

Bits	Description
7:0	<b>DataIndex</b> . Read-write. Reset: 0. It is mapped to 72 data registers in data buffer for block write/read and block write-block read-process call.

**ASFx08 PEC**

Bits	Description
7:0	<b>Pec</b> . Read-write. Reset: 0. PEC byte to be sent to slave.

**ASFx09 ListenAdr**

Bits	Description
7:1	<b>ListenAdr</b> . Read-write. Reset: 0. The slave address which ASF slave responds in listen mode.
0	<b>ListenAdrEn</b> . Read-write. Reset: 0. 1=Enable Listen Mode when the slave address equals to ListenAdr[7:1]. 0=Disable Listen Mode when the slave address equals to ListenAdr[7:1].

**ASFx0A ASFStatus**

Bits	Description
7:6	<b>SlaveBusy</b> . Read-Only. Reset: 0. Indicates if ASF slave is receiving data.
5	<b>SlaveIntr</b> . Read; write-1-to-clear. Reset: 0. ASF Slave interrupt Status.
4	Reserved.
3	<b>RemotePowerCycle</b> . Read; write-1-to-clear. Reset: 0. 1=Power cycle has been triggered by ASF. 0=No power cycle ASF event.
2	<b>RemotePowerUp</b> . Read; write-1-to-clear. Reset: 0. 1=Power up has been triggered by ASF. 0=No Power up ASF event.
1	<b>RemotePowerDown</b> . Read; write-1-to-clear. Reset: 0. 1=Power down has been triggered by ASF. 0=No Power down ASF event.
0	<b>RemoteReset</b> . Read; write-1-to-clear. Reset: 0. 1=Reset has been triggered by ASF. 0=No Reset cycle ASF event.

**ASFx0B StatusMask0**

Bits	Description
7:6	Reserved.
5	<b>FanSpeed0StatusEnable</b> . Read-write. Reset: 0. 1=Report Fan0 Speed Status to ASF. 0=No report.
4:1	Reserved.
0	<b>Temp0StatusEnable</b> . Read-write. Reset: 0. 1=Report TempTsi status to ASF. 0=No report.

**ASFx0C StatusMask1**

Bits	Description
7:0	Reserved.

**ASFx0D SlaveStatus**

Bits	Description
7:4	Reserved.
3	<b>WrongSP.</b> Read; write-1-to-clear. Reset: 0. 1=No SP symbol is detected when bus turns to read.
2	<b>SlaveDevError.</b> Read; write-1-to-clear. Reset: 0. 1=Unexpected response.
1	<b>SlaveBusCollision.</b> Read; write-1-to-clear. Reset: 0. 1=BusCollision happens.
0	<b>SlavePECErrror.</b> Read; write-1-to-clear. Reset: 0. 1=PEC error.

**ASFx0E RemoteCtrlAdr**

Bits	Description
7:1	<b>RemoteCtrlAdr.</b> Read-write. Reset: 54h. SM address of Remote Control device.
0	Reserved.

**ASFx0F SensorAdr**

Bits	Description
7:1	<b>SensorAdr.</b> Read-write. Reset: 55h. SM address of Sensor.
0	Reserved.

**ASFx10 DataReadPointer**

Bits	Description
7:0	<b>DataReadPointer.</b> Read-only. Reset: 0. Current read pointer to the value specified in this register.

**ASFx11 DataWritePointer**

Bits	Description
7:0	<b>DataWritePointer.</b> Read-only. Reset: 0. Show current write pointer to the value specified in this register.



**ASFx12 SetDataReadPointer**

Bits	Description
7:0	<b>SetDataReadPointer</b> . Read-write. Reset: 0. Force the current write pointer to the value specified in this register.

**ASFx13 DataBankSel**

Bits	Description										
7	<b>SetReadHostDataBank</b> . Read-write. Reset: 0. 1=Select to read data from Host Data Bank. 0=Select to read data from Data Bank 0 or Data Bank 1 decided by SetReadRevDataBank.										
6	Reserved.										
5:4	<b>SetReadRevDataBank</b> . Read-write. Reset: 0. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Select to read data from Data Bank 0.</td> </tr> <tr> <td>01b</td> <td>Select to read data from Data Bank 1.</td> </tr> <tr> <td>10b</td> <td>Select to read data from Data Bank 1.</td> </tr> <tr> <td>11b</td> <td>Select to read data from Data Bank 1.</td> </tr> </tbody> </table>	Bits	Definition	00b	Select to read data from Data Bank 0.	01b	Select to read data from Data Bank 1.	10b	Select to read data from Data Bank 1.	11b	Select to read data from Data Bank 1.
Bits	Definition										
00b	Select to read data from Data Bank 0.										
01b	Select to read data from Data Bank 1.										
10b	Select to read data from Data Bank 1.										
11b	Select to read data from Data Bank 1.										
3	<b>Databank1Full</b> . Read; write-1-to-clear. Reset: 0. 0=Data Bank 1 is free. 1=Data Bank 1 is full.										
2	<b>Databank0Full</b> . Read; write-1-to-clear. Reset: 0. 0=Data Bank 0 is free. 1=Data Bank 0 is full.										
1	<b>DataBank[1]</b> . Read; write-1-to-clear. Reset: 0. 0=Data Bank still has space. 1=Data Bank is now full.										
0	<b>DataBank[0]</b> . Read; write-1-to-clear. Reset: 0. 0=Data Bank 0 is the latest touched data bank. 1=Data Bank 1 is the latest touched data bank.										

**ASFx14 Semaphore**

Bits	Description
7:4	Reserved.
3	<b>ClrEcSemaphore</b> . Write-1-only; cleared-by-hardware. Reset: 0. Write 1 to clear [EcSemaphore] bit.
2	<b>EcSemaphore</b> . Read; write-1-only. Reset: 0. This bit can only be set when [HostSemaphore] is clear.
1	<b>ClrHostSemaphore</b> . Write-1-only; cleared-by-hardware. Reset: 0. Write 1 to clear [HostSemaphore] bit.
0	<b>HostSemaphore</b> . Read; write-1-only. Reset: 0. Bits [0] and [2] are meant to be used as software semaphore between the host and IMC. When both host and IMC want to use the same resource, they can write to these semaphore bits first, followed by a read. If the read returns a 1 in the semaphore bit, it means it has established the semaphore first. Write 1 to set this bit. This bit can only be set when EcSemaphore is clear. Writing 0 has no effect. Read returns the value of this bit.

**ASFx15 SlaveEn**

Bits	Description
7	<b>FairArbEn.</b> Read-write. Reset: 0. 0=Disable Fair arbiter logic. 1=Enable Fair Arbiter logic, which forces ASF master to give up SMBus for a certain time, specified in register at offset 16h.
6	<b>TmrOutEn.</b> Read-write. Reset: 0. 1=Enable timer out function.
5	<b>LegacySensorEn.</b> Read-write. Reset: 0. 1=Enable Legacy Sensor.
4	<b>KillSlave.</b> Read-write. Reset: 0. Write 1 to reset Slave ASF Slave state machine.
3	<b>SuspendSlave.</b> Read-write. Reset: 0. Write 1 to Suspend (stop) ASF Slave state machine.
2	Reserved.
1	<b>SlaveIntrListenEn.</b> Read-write. Reset: 0. 1=Allow ASF slave to generate slave interrupt when the address of received packet is the same as the one specified in ListenAdr register <a href="#">ASFx09 [ListenAdr]</a> .
0	Reserved.

**ASFx16 DelayMasterTimer**

Bits	Description
7:0	<b>FairArbTimer.</b> Read-write. Reset: 10h. Specify how long ASF master has to wait before submitting next packet. Wait time = FairArbTimer* 2 us.

**3.25.7.3 SMBus Registers**

SMBus register space is accessed through two methods:

- IO access through SMBus IO base address defined by [PMx2C \[Smbus0En\]](#):
  - Program the base address of SMBus IO space through [PMx2C\[SmBus0IoBase\]](#).
  - Enable SMBus function and IO decoding through [PMx2C\[SmBus0En\]](#).
- Direct memory mapped or IO mapped access through the AcpiMmio region. The SMBus registers range from [PMx24\[AcpiMmioAddr\]+A00h](#) to [PMx24\[AcpiMmioAddr\]+AFFh](#). See [PMx24 \[AcpiMmioEn\]](#).

**SMBUSx00 SMBusStatus**

Bits	Description
7:5	Reserved.
4	<b>Failed.</b> Write-1-to-clear; set-by-hardware; read. Reset: 0. 1=A Failed bus transaction. IF ( <a href="#">SMBUSx02[Kill]</a> ==1) THEN <a href="#">SMBUSx00[Failed]</a> =1. ENDIF.
3	<b>BusCollision.</b> Write-1-to-clear; set-by-hardware; read. Reset: 0. 1=An SMBus transaction collision.
2	<b>DeviceErr.</b> Write-1-to-clear; set-by-hardware; read. Reset: 0. 1=An error of one of the following: <ul style="list-style-type: none"> <li>Illegal command field.</li> <li>Unclaimed cycle</li> <li>Host device time-out.</li> </ul>
1	<b>SMBusInterrupt.</b> Write-1-to-clear; set-by-hardware; read. Reset: 0. 1=The completion of the last host command.
0	<b>HostBusy.</b> Read-only. Reset: 0. 1=The SMBus controller is in the process of completing a command. When this bit is set, software should not access any other SMBus registers.

**SMBUSx01 SMBusSlaveStatus**

Bits	Description
7:6	Reserved.
5	<b>AlertStatus.</b> Read-only. Reset: 0.
4	<b>Shadow2Status.</b> Write-1-to-clear; set-by-hardware; read. Reset: 0. This bit is set by hardware to indicate a slave cycle address match of the SMB_Shadow2 port.
3	<b>Shadow1Status.</b> Write-1-to-clear; set-by-hardware; read. Reset: 0. This bit is set by hardware to indicate a slave cycle address match of the SMB_Shadow1 port.
2	<b>SlaveStatus.</b> Write-1-to-clear; set-by-hardware; read. Reset: 0. This bit is set by hardware to indicate a slave cycle event match of the SMBus slave command and SMBus Slave Event match.
1	<b>SlaveInit.</b> RAZ; write-1-only. Reset: 0. Writing a 1 to this bit initializes the slave.
0	<b>SlaveBusy.</b> Read-only. Reset: 0. This bit indicates the SMBus controller slave interface is in the process of receiving data. Software should not try to access any other SMBus register when this bit is set.

**SMBUSx02 SMBusControl**

Bits	Description																
7	<b>Reset.</b> Write-only. Reset: 0. Set the bit to 1 to stop SMBus transaction and reset SMBus controller state machine.																
6	<b>Start.</b> Read-write. Reset: 0. Writing a 1 in this field initiates SMBus controller host interface to execute the command programmed in the [SMBusProtocol] field.																
5	Reserved.																
4:2	<b>SMBusProtocol.</b> Read-write. Reset: 0. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>Quick Read or Write</td> </tr> <tr> <td>001b</td> <td>Byte Read or Write</td> </tr> <tr> <td>010b</td> <td>Byte Data Read or Write</td> </tr> <tr> <td>011b</td> <td>Word Data Read or Write</td> </tr> <tr> <td>100b</td> <td>Reserved</td> </tr> <tr> <td>101b</td> <td>Block Read or Write</td> </tr> <tr> <td>111b-110b</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Definition	000b	Quick Read or Write	001b	Byte Read or Write	010b	Byte Data Read or Write	011b	Word Data Read or Write	100b	Reserved	101b	Block Read or Write	111b-110b	Reserved
Bits	Definition																
000b	Quick Read or Write																
001b	Byte Read or Write																
010b	Byte Data Read or Write																
011b	Word Data Read or Write																
100b	Reserved																
101b	Block Read or Write																
111b-110b	Reserved																
1	<b>Kill.</b> Read-write. Reset: 0. Stop the current host transaction in process.																
0	<b>InterruptEnable.</b> Read-write. Reset: 0. 1=Enable the generation of interrupts on the completion of current host transaction.																

**SMBUSx03 SMBusHostCmd**

Bits	Description
7:0	<b>SMBusHostCmd.</b> Read-write. Reset: 0. This field contains the data transmitted in the command field of SMBus host transaction.

**SMBUSx04 SMBusAddress**

Bits	Description
7:1	<b>SMBusAddr</b> . Read-write. Reset: 0. This field contains the 7-bit address of the target slave device.
0	<b>SMBusRdWr</b> . Read-write. Reset: 0. 0=Execute a Write command. 1=Execute a Read command.

**SMBUSx05 SMBusData0**

Bits	Description
7:0	<b>SMBusData0</b> . Read-write. Reset: 0. This register should be programmed with a value to be transmitted in the data 0 field of an SMBus host interface transaction. For Block Write commands, the count of the memory should be stored in this field. The value of this register is loaded into the block transfer count field. A valid value for block command count is between 1 and 32. For block reads, count received from SMBus device is stored here.

**SMBUSx06 SMBusData1**

Bits	Description
7:0	<b>SMBusData1</b> . Read-write. Reset: 0. This register should be programmed with a value to be transmitted in the data 1 field of an SMBus host interface transaction. When $(\text{SMBUSx14}[\text{SMBusPoll2Byte}] == 1) \& ((\text{SMBUSx02}[\text{SMBusProtocol}] == \text{Byte Read/Write})   (\text{SMBUSx02}[\text{SMBusProtocol}] == \text{Byte Data Read/Write}))$ , this register is used as the data field in the second transaction.

**SMBUSx07 SMBusBlockData**

Bits	Description
7:0	<b>SMBusBlockData</b> . Read-write. Reset: 0. This register is used to transfer data into or out of the block data storage array.

**SMBUSx08 SMBusSlaveControl**

Bits	Description
7	<b>ClrEcSemaphore</b> . Read-write; Cleared-by-hardware. Reset: 0. Write 1 to clear EcSemaphore bit. Writing 0 has no effect.
6	<b>EcSemaphore</b> . Read-write. Reset: 0. Write 1 to set this bit. This bit can only be set when HostSemaphore is clear. Writing 0 has no effect. Reading returns the value of this bit
5	<b>ClrHostSemaphore</b> . Read-write; Cleared-by-hardware. Reset: 0. Write 1 to clear HostSemaphore bit. Writing 0 has no effect.

4	<b>HostSemaphore.</b> Read-write. Reset: 0. Bits 4 and 6 are meant to be used as software semaphore between the host and embedded controller. When both host and IMC want to use the same resource, they can write to these semaphore bits first, followed by a read. If the read returns a 1 in the semaphore bit, it means it has established the semaphore first. Write 1 to set this bit. This bit can only be set when EcSemaphore is clear. Writing 0 has no effect. Reading returns the value of this bit.
3	<b>SMBusAlertEnable.</b> Read-only. Reset: 0.
2	<b>SMBusShadow2En.</b> Read-write. Reset: 0. Enable the generation of an interrupt or resume event upon an external SMBus master generating a transaction with an address that matches the SMBus Shadow 2 register.
1	<b>SMBusShadow1En.</b> Read-write. Reset: 0. Enable the generation of an interrupt or resume event upon an external SMBus master generating a transaction with an address that matches the SMBus Shadow 1 register.
0	<b>SlaveEnable.</b> Read-write. Reset: 0. Enable the generation of an interrupt or resume event upon an external SMBus master generating a transaction with an address that matches the host controller slave port of 10h, a command field that matches the SMBus slave control register, and a match of corresponding enabled events.

### SMBUSx09 SMBusShadowCmd

Bits	Description
7:0	<b>SMBusShadowCmd.</b> Read-write. Reset: 0. This field contains the command value that was received during an external SMBus master access whose address field matched the host slave address (10h) or one of the slave shadow ports.

### SMBUSx0A SMBusSlaveEvent

Bits	Description
15:0	<b>SMBusSlaveEvent.</b> Read-write. Reset: 0. This field contains data bits used to compare against incoming data to the SMBus Slave Data register. When a bit in this register is 1 and a corresponding bit in SMBus Slave register is set, then an interrupt or resume event is generated if the command value matches the value in the SMBus slave control register and the access was to SMBus host address 10h.

### SMBUSx0C SlaveData

Bits	Description
15:0	<b>SlaveData.</b> Read-write. Reset: 0. This field contains the data value which was transmitted during an external SMBus master access whose address field matched one of the slave shadow port addresses or the SMBus host controller slave port address of 10h.

**SMBUSx0E SMBusTiming**

Bits	Description
7:0	<b>SMBusTiming</b> . Read-write. Reset: B0h. This register controls the frequency on the SMBUS. The formula to calculate the frequency is: Frequency = 66 MHz/(SMBusTiming * 4).

**SMBUSx10 I2CbusConfig**

Bits	Description
7:4	<b>I2CRevision</b> . Read-write. Reset: 0. Sm bus controller revision.
3:1	Reserved.
0	<b>I2CbusInterrupt</b> . Read-write. Reset: 0. 0=SMI#. 1=IRQ.

**SMBUSx11 I2CCommand**

Bits	Description
7:0	<b>I2Ccommand</b> . Read-write. Reset: 0. I2C Host Slave Command. This value specifies the command value to be matched for I2C master accesses to the I2Ccontroller host slave interface.

**SMBUSx12 I2CShadow1**

Bits	Description
7:1	<b>I2CslaveAddr1</b> . Read-write. Reset: 0. SMBus Slave Address for shadow port 1. This value specifies the address used to match against incoming I2C addresses for Shadow port 1.
0	<b>ReadWriteShadowPort1</b> . Read-write. Reset: 0. This bit must be programmed to 0 because I2C slave controller only responds to Word Write Transaction. 0=Write command. 1=Read command.

**SMBUSx13 I2Cshadow2**

Bits	Description
7:1	<b>I2CslaveAddr2</b> . Read-write. Reset: 0. SMBus Slave Address for shadow port 2. This value specifies the address used to match against incoming I2C addresses for Shadow port 2.
0	<b>ReadWriteShadowPort2</b> . Read-write. Reset: 0. This bit must be programmed to 0 because I2C slave controller only responds to Word Write Transaction. 0=Write command. 1=Read command.

**SMBUSx14 SMBusAutoPoll**

Bits	Description												
7	<b>SMBusPoll2Byte.</b> Read-write. Reset: 1. When set, if ((SMBUSx02[SMBusProtocol]==Byte Read/Write)   (SMBUSx02[SMBusProtocol]==Byte Data Read/Write), the last command will be performed twice at each polling interval. SMBUSx06 [SMBusData1] contains data transmitted and retrieved from the second transaction. SMBUSx17 [SMBusHostCmd2] contains data transmitted in the command field in the second transaction.												
6:4	Reserved.												
3:1	<b>SMBusPollPeriod.</b> Read-write. Reset: 2h. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>poll every 1/16 second</td> </tr> <tr> <td>001b</td> <td>poll every 1/8 second</td> </tr> <tr> <td>010b</td> <td>poll every 1/4 second</td> </tr> <tr> <td>011b</td> <td>poll every 1/2 second</td> </tr> <tr> <td>111b-100b</td> <td>poll every 1 second</td> </tr> </tbody> </table>	Bits	Definition	000b	poll every 1/16 second	001b	poll every 1/8 second	010b	poll every 1/4 second	011b	poll every 1/2 second	111b-100b	poll every 1 second
Bits	Definition												
000b	poll every 1/16 second												
001b	poll every 1/8 second												
010b	poll every 1/4 second												
011b	poll every 1/2 second												
111b-100b	poll every 1 second												
0	<b>SMBusAutoPollEn.</b> Read-write. Reset: 0. 1=SMBUS will periodically execute the last command whenever SMBUSx15 [SMBusCounter] expires. The polling interval is defined by [SMBusPollPeriod]. The purpose of this function is to use the SMBUS to read CPU temperature via TSI. Values in SMBUSx05 [SMBusData0] and SMBUSx06 [SMBusData1] are to be used for controlling the fan speed.												

**SMBUSx15 SMBusCounter**

Bits	Description
7:0	<b>SMBusCounter.</b> Read-only. This counter specifies the remaining time until the next read. The interval is based on SMBUSx14[SMBusPollPeriod]. For example, if SMBUSx14[SMBusPollPeriod] is 000b, then each tick in this counter represents $1/16 / 256 = 244\mu\text{s}$ .

**SMBUSx16 SMBusStop**

Bits	Description
15:1	Reserved.
0	<b>SMBusPausePoll.</b> Read-write. Reset: 0. This is to be used as a semaphore mechanism by software to stop the next polling. If software wants to use the SMBUS for other purpose but the hardware has already been enabled (SMBUSx14[SMBusAutoPollEn]=1) to perform auto polling, software can halt the next polling by setting this bit to 1. If the halt is successful, this bit will return 1. If the HW has already started the polling operation at the same time software tries to set this bit, this bit will return 0. When the pending polling operation is finally complete, this bit will then get set. When this bit is set, polling counter will temporarily stop (not reset). It is now safe for software to access the SMBUS. Upon completion, software must restore the previous command in the appropriate register prior to setting this bit. software should also clear this bit so the counter can resume counting.

**SMBUSx17 SMBusHostCmd2**

---

Bits	Description
7:0	<b>SMBusHostCmd2</b> . Read-write. Reset: 0. This field contains the data transmitted in the command field in the second command when ( <b>SMBUSx14</b> [SMBusPoll2Byte]==1) and ( <b>SMBUSx02</b> [SMBus-Protocol]==Byte Data Read/Write).



### 3.25.8 IOAPIC Registers

IOAPIC configuration registers are accessed through IOAPIC base address defined by [PMx34 \[IoApicEn\]](#):

1. Select the IOAPIC register space to be memory mapped or IO mapped through [PMx34\[IoApicMmIo\]](#).
2. Program the IOAPIC base address through [PMx34\[IoApicBaseAddr\]](#).
3. Enable IOAPIC decoding through [PMx34\[IoApicEnable\]](#).

#### IOAPICx00 IO Register Select Register

Bits	Description
31:8	Reserved.
7:0	<b>IndirectAddressOffset</b> . Read-write. Reset: 0. Indirect Address Offset to IO Window Register. It is used to determine which register is manipulated during an IO Window Register read/write operation.

#### IOAPICx10 IO Window Register

Bits	Description
31:0	<b>IOWindow</b> . Read-write. Reset: 0. Mapped by <a href="#">IOAPICx00</a> to the designated indirect access register.

#### IOAPICx10\_x00 IOAPIC ID Register

This register is not used in IOxAPIC PCI bus delivery mode.

Bits	Description
31:24	<b>ID</b> . Read-write. Reset: 0. IOAPIC device ID. IF ( <a href="#">PMx34[IoApicIdExtEn]</a> ==1) THEN This field is 8-bit wide. ELSE This field is 8-bit wide and [31:28] are reserved. ENDIF.
23:0	Reserved.

#### IOAPICx10\_x01 IOAPIC Version Register

Bits	Description
31:24	Reserved.
23:16	<b>MaxRedirectionEntries</b> . Value: 17h. Indicates 24 entries [23:0]
15	<b>PRQ</b> . Value: 1. IRQ pin assertion supported
14:8	Reserved.
7:0	<b>Version</b> . Value: 21h. PCI 2.2 compliant

#### IOAPICx10\_x02 IOAPIC Arbitration Register

This register is not used in IOxAPIC PCI bus delivery mode.

Bits	Description
31:28	Reserved.
27:24	<b>ArbitrationID: Arbitration ID</b> . Read-only. Reset: 0.
23:0	Reserved.

**IOAPICx10\_x[3E:10:step2] Redirection Table Entry [23:0]**

Bits	Description																				
63:56	<b>DestinationID.</b> Read-write. Reset: 0. Bits [19:12] of the address field of the interrupt message																				
55:32	Reserved.																				
31:17	Reserved.																				
16	<b>Mask.</b> Read-write. Reset: 1. 1=Mask the interrupt injection at the input of this device. 0=Unmask.																				
15	<b>TriggerMode.</b> Read-write. Reset: 0. 0=Edge. 1=Level																				
14	<b>RemoteIRR.</b> Read-only. Reset: 0. Used for level triggered interrupts only. It is cleared by EOI special cycle transaction or write to EOI register. 1=Interrupt message is delivered.																				
13	<b>InterruptPinPolarity.</b> Read-write. Reset: 0. 0=High. 1=Low.																				
12	<b>DeliveryStatus.</b> Read-only. Reset: 0. 0=Idle. 1=Send Pending.																				
11	<b>DestinationMode.</b> Read-write. Reset: 0. 0=Physical. 1=Logical																				
10:8	<b>DeliveryMode.</b> Read-write. Reset: 0. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;"><u>Bits</u></th> <th style="text-align: left;"><u>Definition</u></th> <th style="text-align: left;"><u>Bits</u></th> <th style="text-align: left;"><u>Definition</u></th> </tr> </thead> <tbody> <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/PMI</td> <td>110b</td> <td>Reserved</td> </tr> <tr> <td>011b</td> <td>Reserved</td> <td>111b</td> <td>ExtINT</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	<u>Bits</u>	<u>Definition</u>	000b	Fixed	100b	NMI	001b	Lowest Priority	101b	INIT	010b	SMI/PMI	110b	Reserved	011b	Reserved	111b	ExtINT
<u>Bits</u>	<u>Definition</u>	<u>Bits</u>	<u>Definition</u>																		
000b	Fixed	100b	NMI																		
001b	Lowest Priority	101b	INIT																		
010b	SMI/PMI	110b	Reserved																		
011b	Reserved	111b	ExtINT																		
7:0	<b>Vector.</b> Read-write. Reset: 0. Interrupt vector associated with this interrupt input																				

**IOAPICx20 IRQ Pin Assertion Register**

Bits	Description
31:8	Reserved.
7:0	<b>InputIrq.</b> Read-write. Reset: 0. IRQ number for the requested interrupt. A write to this register will trigger an interrupt associated with the redirection table entry referenced by the IRQ number. Currently the redirection table has 24 entries. Writes with IRQ number greater than 17h have no effect.

**IOAPICx40 EOI Register**

Bits	Description
31:8	Reserved.
7:0	<b>Vector.</b> Write-only. Reset: 0. Interrupt vector. A write to this register will clear the remote IRR bit in the redirection table entry found matching the interrupt vector. This provides an alternate mechanism other than PCI special cycle for EOI to reach IOxAPIC.

### 3.25.9 LPC-ISA Bridge

#### 3.25.9.1 Device 14h Function 3 (LPC Bridge) 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.

##### D14F3x00 Device/Vendor ID

Bits	Description
31:16	<b>DeviceID: Device Identifier.</b> Value: 780Eh. This 16-bit field is assigned by the device manufacturer and identifies the type of device.
15:0	<b>VendorID: Vendor Identifier.</b> Read-only. Reset: 1022h.

##### D14F3x04 Status/Command

Bits	Description
31	<b>DetectedParityError: Detected Parity Error.</b> Read; write-1-to-clear. Reset: 0. 1=The FCH detects a parity error.
30	<b>SignaledSystemError: Signaled System Error.</b> Read; write-1-to-clear. Reset: 0. 1=The FCH detects a PCI address parity error.
29	<b>ReceivedMasterAbort: Received Master Abort.</b> Read; write-1-to-clear. Reset: 0. 1=The FCH acts as a PCI master and aborts a PCI bus memory cycle.
28	<b>ReceivedTargetAbort: Received Target Abort.</b> Read; write-1-to-clear. Reset: 0. 1=An FCH generated PCI cycle (the FCH is the PCI master) is aborted by a PCI target.
27	<b>SignaledTargetAbort: Signaled Target Abort.</b> Read; write-1-to-clear. Reset: 0. 1=The FCH signals target abort.
26:25	<b>DeviceSelectTiming: Device Select Timing.</b> Value: 1. Indicates DEVSEL# timing when performing a positive decode. DEVSEL# is asserted to meet the medium timing.
24	<b>MasterDataParityError: Master Data Parity Error.</b> Read; write-1-to-clear. Reset: 0. 1=The FCH detects PERR# asserted while acting as PCI master regardless whether PERR# was driven by the FCH or not.
23:21	Reserved.
20	<b>CapabilitiesList: Capabilities List.</b> Read-only. Reset:0.
19:10	Reserved.
9	<b>FastBack2BackEnable: Fast Back-to-Back Enable.</b> Value: 0.
8	<b>SERREnable: SERR# Enable.</b> Read-only. Reset: 0. 1=The FCH asserts SERR# when it detects an address parity error. 0=SERR# is not asserted.
7	<b>SteppingControl: Stepping Control.</b> Value: 0.
6	<b>ParityErrorResponse: Parity Error Response.</b> Read-write. Reset: 0. PERR# (Response) Detection Enable bit. 1=The FCH asserts PERR# when it is the agent receiving data and it detects a parity error. 0=PERR# is not asserted.
5	<b>VGAPaletteSnoop: VGA Palette Snoop.</b> Value: 0.

4	<b>MemoryWriteAndInvalidateEnable: Memory Write and Invalidate Enable.</b> Value: 0.
3	<b>SpecialCycles: Special Cycles.</b> Value: 1.
2	<b>BusMaster: Bus Master.</b> Read-only. Reset: 1. 1=Bus master enabled.
1	<b>MemorySpace: Memory Space.</b> Value: 1.
0	<b>IOspace: IO Space.</b> Read-only. Reset: 1. This bit controls access to the IO space registers. 1=Enable access to the legacy IDE ports, and PCI bus master IDE IO registers are enabled.

#### D14F3x08 Revision ID/Class Code

Bits	Description
31:8	<b>ClassCode: Class Code.</b> Read-only. Reset: 60100h. Indicates an ISA bridge.
7:0	<b>RevisionID: Revision ID.</b> Value: 11h. Indicates the revision level of the chip design.

#### D14F3x0C Cache Line Size

Bits	Description
31:24	<b>Bist.</b> Read-only. Reset: 0. No BIST modes.
23:16	<b>HeaderType: Header Type.</b> Read-only. Reset: 80h. Identifies the type of the predefined header in the configuration space. The most significant bit is set to 1 to indicate a multifunction device.
15:8	<b>LatencyTimer: Latency Timer.</b> Read-only. Reset: 0. Specifies the value of the latency timer in units of PCICLKs.
7:0	<b>CacheLineSize: Cache Line Size.</b> Value: 0.

#### D14F3x10 Base Address Reg 0

Bits	Description
31:5	<b>BaseAddress0[31:5]: Base Address 0.</b> RAZ; write-only. Reset: 7F60000h. This register has an internal value used as base address for APIC memory space. Writing to the register changes its internal value. The default internal base address is FEC0_0000h.
4:0	<b>BaseAddress0[4:0].</b> Value: 0.

#### D14F3x2C Subsystem ID and Subsystem Vendor ID

Bits	Description
31:16	<b>SubsystemID: Subsystem ID.</b> Write-once; Read. Reset: 780Eh.
15:0	<b>SubsystemVendorID: Subsystem Vendor ID.</b> Write-once; Read. Reset: 1022h.

#### D14F3x34 Capabilities Pointer

Bits	Description
------	-------------

31:8	Reserved.
7:0	<b>CapabilitiesPointer: Capabilities Pointer.</b> Read-only. Reset:0.

### D14F3x40 PCI Control

Bits	Description
31:8	Reserved.
7	<b>IntegratedEcPresent.</b> Read-write. Reset: 0. This bit can only be programmed by IMC. 1=IMC is present.
6	<b>EcSemaphore.</b> Read-write. Reset: 0. This bit is writable by IMC and read by BIOS. This is used as the software semaphore mechanism between BIOS and IMC to see who can access the common resource. IMC should read bit [BiosSemaphore] first to see if BIOS has taken ownership of the resource. If [BiosSemaphore] == 0, then IMC should write a 1 to this bit and then follow by a read to see if this bit is set. If this bit is set, it means IMC has successfully taken ownership of the resource. If this bit returns a 0 and [BiosSemaphore] returns a 1, then BIOS has taken ownership first. IMC should always clear this bit after it has completed its access to the resource.
5	<b>BiosSemaphore.</b> Read-write. Reset: 0. This bit is writable by BIOS and read by the IMC. This is used as the software semaphore mechanism between BIOS and IMC to see who can access the common resource. BIOS should read bit [EcSemaphore] first to see if IMC has taken ownership of the resource. If [EcSemaphore] == 0, then BIOS should write a 1 to this bit and then follow by a read to see if this bit is set. If this bit is set, it means BIOS has successfully taken ownership of the resource. If this bit returns 0 and [EcSemaphore] returns a 1, then IMC has taken ownership first. Software should always clear this bit after it has completed its access to the resource.
4:3	Reserved.
2	<b>LegacyDmaEnable: Legacy DMA Enable.</b> Read-write. Reset: 1. BIOS: See 2.15.4.1. 1=Enable LPC DMA cycle. Transfer size for channel 3-0 is 8 bits; Transfer size for channel 7-5 is 16 bits. 32-bit DMA is not supported.
1:0	Reserved.

### D14F3x44 IO Port Decode Enable

Bits	Description
31	<b>AdLibPortEnable: Ad-Lib Port Enable.</b> Read-write. Reset: 0. Port enable for Ad-Lib port, 388h-389h. 1=Enable the IO range.
30	<b>AcpiMicroControllerPortEnable: ACPI Micro-Controller Port Enable.</b> Read-write. Reset: 0. Port enable for ACPI micro-controller port, 62h & 66h. 1=Enable the IO range.
29	<b>KBCPortEnable: KBC Port Enable.</b> Read-write. Reset: 0. Port enable for KBC port, 60h & 64h. 1=Enable the IO range.
28	<b>GamePortEnable: Game Port Enable.</b> Read-write. Reset: 0. Port enable for game port, 200h-20Fh. 1=Enable the IO range.
27	<b>FDCPortEnable1: FDC Port Enable 1.</b> Read-write. Reset: 0. Port enable for FDC port, 370h-377h. 1=Enable the IO range.
26	<b>FDCPortEnable0: FDC Port Enable 0.</b> Read-write. Reset: 0. Port enable for FDC port, 3F0h-3F7h. 1=Enable the IO range.

25	<b>MSSPortEnable3: MSS Port Enable 3.</b> Read-write. Reset: 0. Port enable for MSS port, F40h-F47h. 1=Enable the IO range.
24	<b>MSSPortEnable2: MSS Port Enable 2.</b> Read-write. Reset: 0. Port enable for MSS port, E80h-E87h. 1=Enable the IO range.
23	<b>MSSPortEnable1: MSS Port Enable 1.</b> Read-write. Reset: 0. Port enable for MSS port, 604h-60bh. 1=Enable the IO range.
22	<b>MSSPortEnable0: MSS Port Enable 0.</b> Read-write. Reset: 0. Port enable for MSS port, 530h-537h. 1=Enable the IO range.
21	<b>MIDIPortEnable3: MIDI Port Enable 3.</b> Read-write. Reset: 0. Port enable for MIDI port, 330h-331h. 1=Enable the IO range.
20	<b>MIDIPortEnable2: MIDI Port Enable 2.</b> Read-write. Reset: 0. Port enable for MIDI port, 320h-321h. 1=Enable the IO range.
19	<b>MIDIPortEnable1: MIDI Port Enable 1.</b> Read-write. Reset: 0. Port enable for MIDI port, 310h-311h. 1=Enable the IO range.
18	<b>MIDIPortEnable0: MIDI Port Enable 0.</b> Read-write. Reset: 0. Port enable for MIDI port, 300h-301h. 1=Enable the IO range.
17	<b>AudioPortEnable3: Audio Port Enable 3.</b> Read-write. Reset: 0. Port enable for audio port, 280h-293h. 1=Enable the IO range.
16	<b>AudioPortEnable2: Audio Port Enable 2.</b> Read-write. Reset: 0. Port enable for audio port, 260h-273h. 1=Enable the IO range.
15	<b>AudioPortEnable1: Audio Port Enable 1.</b> Read-write. Reset: 0. Port enable for audio port, 240h-253h. 1=Enable the IO range.
14	<b>AudioPortEnable0: Audio Port Enable 0.</b> Read-write. Reset: 0. Port enable for audio port, 230h-233h. 1=Enable the IO range.
13	<b>SerialPortEnable7: Serial Port Enable 7.</b> Read-write. Reset: 0. Port enable for serial port, 3E8h-3EFh. 1=Enable the IO range.
12	<b>SerialPortEnable6: Serial Port Enable 6.</b> Read-write. Reset: 0. Port enable for serial port, 338h-33Fh. 1=Enable the IO range.
11	<b>SerialPortEnable5: Serial Port Enable 5.</b> Read-write. Reset: 0. Port enable for serial port, 2E8h-2EFh. 1=Enable the IO range.
10	<b>SerialPortEnable4: Serial Port Enable 4.</b> Read-write. Reset: 0. Port enable for serial port, 238h-23Fh. 1=Enable the IO range.
9	<b>SerialPortEnable3: Serial Port Enable 3.</b> Read-write. Reset: 0. Port enable for serial port, 228h-22Fh. 1=Enable the IO range.
8	<b>SerialPortEnable2: Serial Port Enable 2.</b> Read-write. Reset: 0. Port enable for serial port, 220h-227h. 1=Enable the IO range.
7	<b>SerialPortEnable1: Serial Port Enable 1.</b> Read-write. Reset: 0. Port enable for serial port, 2F8h-2FFh. 1=Enable the IO range.
6	<b>SerialPortEnable0: Serial Port Enable 0.</b> Read-write. Reset: 0. Port enable for serial port, 3F8h-3FFh. 1=Enable the IO range.
5	<b>ParallelPortEnable5: Parallel Port Enable 5.</b> Read-write. Reset: 0. Port enable for parallel port, 7BCh-7BFh. 1=Enable the IO range.
4	<b>ParallelPortEnable4: Parallel Port Enable 4.</b> Read-write. Reset: 0. Port enable for parallel port, 3BCh-3BFh. 1=Enable the IO range.

3	<b>ParallelPortEnable3: Parallel Port Enable 3.</b> Read-write. Reset: 0. Port enable for parallel port, 678h-67Fh. 1=Enable the IO range.
2	<b>ParallelPortEnable2: Parallel Port Enable 2.</b> Read-write. Reset: 0. Port enable for parallel port, 278h-27Fh. 1=Enable the IO range.
1	<b>ParallelPortEnable1: Parallel Port Enable 1.</b> Read-write. Reset: 0. Port enable for parallel port, 778h-77Fh. 1=Enable the IO range.
0	<b>ParallelPortEnable0: Parallel Port Enable 0.</b> Read-write. Reset: 0. Port enable for parallel port, 378h-37Fh. 1=Enable the IO range.

### D14F3x48 IO/Mem Port Decode Enable

Bits	Description
31:26	Reserved.
25	<b>WideIO2Enable.</b> Read-write. Reset: 0. Port enable for wide generic IO port 2 defined by <a href="#">D14F3x90</a> [IOBaseAddress2]. 1=Enable the IO range.
24	<b>WideIO1Enable.</b> Read-write. Reset: 0. Port enable for wide generic IO port 1 defined by <a href="#">D14F3x64</a> [IOBaseAddress1]. 1=Enable the IO range.
23	<b>IOPortEnable6: IO port enable 6.</b> Read-write. Reset: 0. Port enable for IO port FD60h-FD6Fh. 1=Enable the IO range.
22	<b>IOPortEnable5: IO port enable 5.</b> Read-write. Reset: 0. Port enable for IO port 4700h-470Bh. 1=Enable the IO range.
21	<b>IOPortEnable4: IO port enable 4.</b> Read-write. Reset: <u>1</u> . Port enable for IO port 80h. 1=Enable the IO range.
20	<b>MemPortEnable: Mem port enable.</b> Read-write. Reset: 0. Port enable for 4K byte memory range defined in <a href="#">D14F3x4C</a> [Memory Range]. 1=Enable the memory range.
19	<b>IOPortEnable3: IO port enable 3.</b> Read-write. Reset: 0. Port enable for IO port 580h-5BFh. 1=Enable the IO range.
18	<b>IOPortEnable2: IO port enable 2.</b> Read-write. Reset: 0. Port enable for IO port 500h-53Fh. 1=Enable the IO range.
17	<b>IOPortEnable1: IO port enable 1.</b> Read-write. Reset: 0. Port enable for IO port 480h-4BFh. 1=Enable the IO range.
16	<b>IOPortEnable0: IO port enable 0.</b> Read-write. Reset: 0. Port enable for IO port 400h-43Fh. 1=Enable the IO range.
15:8	<b>SyncTimeoutCount: Sync Timeout Count.</b> Read-write. Reset: FFh. When [SyncTimeoutCounterEnable] == 1, this is the number of LPC clocks that the state machine will wait during LPC data sync before aborting the cycle.
7	<b>SyncTimeoutCounterEnable: Sync Timeout Counter Enable.</b> Read-write. Reset: 0. 1=LPC sync timeout counter is enabled. 0=The counter is disabled. This counter is used to avoid a deadlock condition if an LPC device drives sync forever. Timeout count is programmed in [SyncTimeoutCount]. Write 0 to this bit if an LPC device is extremely slow and takes more than 255 LPC clocks to complete a cycle.
6	<b>RtcIORangePortEnable: RTC IO Range Port Enable.</b> Read-write. Reset: 0. Port enable for RTC IO range 70h-73h. 1=Enable the IO range.

5	<b>MemoryRangePortEnable: Memory Range Port Enable.</b> Read-write. Reset: 0. Port enable for LPC memory target range defined by <a href="#">D14F3x60 [PCI Memory Address for LPC Target Cycles]</a> . 1=Enable the memory range.
4	<b>RomRange2PortEnable: ROM Range 2 Port Enable.</b> Read-write. Reset: 0. Port enable for LPC ROM address range 2 defined by <a href="#">D14F3x6C [ROM Address Range 2]</a> . 1=Enable the ROM range.
3	<b>RomRange1PortEnable: ROM Range 1 Port Enable.</b> Read-write. Reset: 0. Port enable for LPC ROM address range 1 defined by <a href="#">D14F3x68 [ROM Address Range 1]</a> . 1=Enable the ROM range.
2	<b>WideIO0Enable.</b> Read-write. Reset: 0. Port enable for wide generic IO port defined by <a href="#">D14F3x64[IOBaseAddress0]</a> . 1=Enable the IO range.
1	<b>AlternateSuperIOConfigurationPortEnable: Alternate Super IO Configuration Port Enable.</b> Read-write. Reset: 0. Port enable for alternate Super IO configuration port, 4Eh-4Fh. 1=Enable the IO range.
0	<b>SuperIOConfigurationPortEnable: Super IO Configuration Port Enable.</b> Read-write. Reset: 0. Port enable for Super IO configuration port, 2Eh-2Fh. 1=Enable the IO range.

### D14F3x4C Memory Range

Bits	Description
31:12	<b>BaseAddress: Base Address.</b> Read-write. Reset: 0. Specifies a 4K byte memory range from {Base Address, 000h} to {Base Address, FFFh}. The range is enabled by <a href="#">D14F3x48[MemPortEnable]</a> .
11:0	Reserved.

### D14F3x[5C,58,54,50] ROM Protect 3, 2, 1, 0

These registers specify different ROM ranges to be protected. [SPIx1D\[SpiProtectEn0\]](#) enables the protection ranges. The addresses are within the defined ROM range if:

$$\{\text{RomBase}, 000\_0000\_0000b\} \leq \text{address}[31:0] \leq (\{\text{RomBase}, 000\_0000\_0000b\} + \{0\_0000\_0000\_0000b, \text{RomOffset}, 11\_1111\_1111b\}).$$

For host, these registers can only be written once after hardware reset; Subsequent writes to it have no effect. For IMC, these registers are always read-writable.

Bits	Description
31:11	<b>RomBase: ROM Base.</b> Read; write-once. Reset: 0.
10:2	<b>RomOffset: ROM Offset.</b> Read; write-once. Reset: 0.
1	<b>ReadProtect: Read Protect.</b> Read; write-once. Reset: 0. 1=The memory range defined by this register is read-protected and reading any location in the range returns FFFF_FFFFh.
0	<b>WriteProtect: Write Protect.</b> Read; write-once. Reset: 0. 1=The memory range defined by this register is write-protected and writing to the range has no effect.

### D14F3x60 PCI Memory Address for LPC Target Cycles

This register contains the upper 16 bits of the start and end address of the LPC memory target range. The lower 16 bits of the start address are considered 0s. The lower 16 bits of the end address are considered 1s. This range can be enabled or disabled using [D14F3x48\[MemoryRangePortEnable\]](#).



Bits	Description
31:16	<b>MemoryEndAddress: Memory End Address.</b> Read-write. Reset: 0. Specifies the upper 16 bits of the end address of the LPC target memory range.
15:0	<b>MemoryStartAddress: Memory Start Address.</b> Read-write. Reset: 0. Specifies the upper 16 bits of the start address of the LPC target memory range.

### D14F3x64 PCI IO base Address for Wide Generic Port

Bits	Description
31:16	<b>IOBaseAddress1: IO Base Address 1.</b> Read-write. Reset: 0. 16-bit PCI IO base address for wide generic IO port range. This function is enabled by <a href="#">D14F3x48[WideIO1Enable]</a> . If <a href="#">D14F3x74[AlternativeWideIO1RangeEnable]</a> == 1, the range is 16 bytes; else the range is 512 bytes.
15:0	<b>IOBaseAddress0: IO Base Address 0.</b> Read-write. Reset: 0. 16-bit PCI IO base address for wide generic IO port range. This function is enabled by <a href="#">D14F3x48[WideIO0Enable]</a> . If <a href="#">D14F3x74[AlternativeWideIO0RangeEnable]</a> == 1, the range is 16 bytes; else the range is 512 bytes.

### D14F3x68 ROM Address Range 1

This register contains the upper 16 bits of the start and end address of the ROM address range 1. The lower 16 bits of the start address are considered 0s. The lower 16 bits of the end address are considered 1s. This range can be enabled or disabled using [D14F3x48\[RomRange1PortEnable\]](#). This register is used for both LPC and SPI flash.

Bits	Description
31:16	<b>RomEndAddress1: ROM End Address 1.</b> Read-write. Reset: 0Fh. Specifies the upper 16 bits of the end address of the ROM memory address range 1.
15:0	<b>RomStartAddress1: ROM Start Address 1.</b> Read-write. Reset: 08h. Specifies the upper 16 bits of the start address of the ROM memory address range 1.

### D14F3x6C ROM Address Range 2

This register contains the upper 16 bits of the start and end address of the ROM address range 2. The lower 16 bits of the start address are considered 0s. The lower 16 bits of the end address are considered 1s. This range can be enabled or disabled using [D14F3x48\[RomRange2PortEnable\]](#). This register is used for both LPC and SPI flash.

Bits	Description
31:16	<b>RomEndAddress2: ROM End Address 2.</b> Read-write. Reset: FFFFh. Specifies the upper 16 bits of the end address of the ROM memory address range 2.
15:0	<b>RomStartAddress2: ROM Start Address 2.</b> Read-write. Reset: FFF8h. Specifies the upper 16 bits of the start address of the ROM memory address range 2.

**D14F3x74 Alternative Wide IO Range Enable**

Bits	Description
31:4	Reserved.
3	<b>AlternativeWideIO2RangeEnable: Alternative Wide IO 2 Range Enable.</b> Read-write. Reset: 0. This bit is similar to bit [AlternativeWideIO0RangeEnable], but it applies to the IO range defined by <a href="#">D14F3x90</a> [IOBaseAddress2]. See bit[AlternativeWideIO0RangeEnable] for detailed description.
2	<b>AlternativeWideIO1RangeEnable: Alternative Wide IO 1 Range Enable.</b> Read-write. Reset: 0. This bit is similar to bit [AlternativeWideIO0RangeEnable], but it applies to the IO range defined by <a href="#">D14F3x64</a> [IOBaseAddress1]. See bit [AlternativeWideIO0RangeEnable] for detailed description.
1	Reserved.
0	<b>AlternativeWideIO0RangeEnable: Alternative Wide IO 0 Range Enable.</b> Read-write. Reset: 0. 0=Wide IO range defined by <a href="#">D14F3x64</a> [IOBaseAddress0] is 512 bytes. 1=The range is 16 bytes. To use this feature, address in <a href="#">D14F3x64</a> [IOBaseAddress0] must be aligned to 16 bytes, i.e., bits[3:0] must be 0. If the address is not aligned to 16 bytes, the IO range is from address[15:0] to {address[15:4], 0xF}.

**D14F3x78 Miscellaneous Control Bits**

Bits	Description
31:9	Reserved.
8	Reserved.
7	<b>AllowHostInDma.</b> Read-write. Reset: 1. 1=Allow Host to access LPC if ACPI has not given GNT to LPC during DMA transfer. 0=DMA hold LPC even ACPI has not given GNT to LPC during DMA transfer.
6	<b>GateWrongRx.</b> Read-write. Reset: 0. 1=Allow AltRxBByteCount to be 0.
5	<b>GateSpiAccessDis.</b> Read-write. Reset: 0. 1=Pass ROM access to SPI even if it is strapped as LPC.
4	<b>SMMWriteRomEn.</b> Read-write. Reset: 1. 1=Enable ROM access in SMM mode.
3	<b>LDRQ1.</b> Read-write. Reset: 0. BIOS: See <a href="#">2.15.4.1</a> . 1=Enable LDRQ1# on LPC bus.
2	<b>LDRQ0.</b> Read-write. Reset: 0. BIOS: See <a href="#">2.15.4.1</a> . 1=Enable LDRQ0# on LPC bus.
1	Reserved.
0	<b>NoHog: No Hog.</b> Read-write. Reset: 1. BIOS: See <a href="#">2.15.4.1</a> . 1=The internal bus is not locked by LPC bridge during a slave access (eg. LPC DMA fetch). 0=LPC may hold the internal bus during a DMA transfer.

**D14F3x7C TPM**

Bits	Description
31:14	Reserved.
13	<b>TpmBufferEn.</b> Read-write. Reset: 0. 1=Enable TPM buffer. 0=Disable TPM buffer.
12	<b>TpmPfetchEn.</b> Read-write. Reset: 0. 1=Enable TPM burst read. 0=Disable TPM burst read.
11	<b>LpcClk1IsGpio.</b> Read-write. Reset: 1. 1=Treat LpcClk1 as GPIO. 0=Treat LpcClk1 as LpcClk1.

10	<b>GpioLpcClk1Out.</b> Read-write. Reset: 0. Control GpioLpcClk1 output value.										
9	<b>GpioLpcClk1OeB.</b> Read-write. Reset: 1. 1=disable GpioLpcClk1 output. 0=Enable GpioLpcClk1 output.										
8	<b>GpioLpcClk1.</b> Read-only. Reset: X. Status of LpcClk1 port.										
7	<b>WiderTpmEn.</b> Read-write. Reset: 0. 1=Force logic to decode FED4_xxxxh as TPM cycles instead of FED4_0xxxh, FED4_1xxxh, FED4_2xxxh, FED4_3xxxh, and FED4_4xxxh.										
6:5	<p><b>TmkbcSel.</b> Read-write. Reset: 0. Select which one of the four sets of TMKBC registers specified in <a href="#">D14F3x84</a>, <a href="#">D14F3x88</a>, <a href="#">D14F3x8C</a> are accessed.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>set 0</td> </tr> <tr> <td>01b</td> <td>set 1</td> </tr> <tr> <td>10b</td> <td>set 2</td> </tr> <tr> <td>11b</td> <td>set 3</td> </tr> </tbody> </table>	Bits	Definition	00b	set 0	01b	set 1	10b	set 2	11b	set 3
Bits	Definition										
00b	set 0										
01b	set 1										
10b	set 2										
11b	set 3										
4	<b>TmkbcSet.</b> Write-once. Reset: 0. 1=All TMKBC address/remap registers cannot be changed until the next reset.										
3	<b>TmkbcEnable.</b> Read-write. Reset: 0. 1=Enable the TMKBC function.										
2	<b>TpmLegacy.</b> Read-write. Reset: 0. 1=Enable decoding of legacy TPM addresses: IO addresses 7Eh/7Fh and EEh/EFh.										
1	Reserved.										
0	<b>Tpm12En.</b> Read-write. Reset: 0. 1=Enable decoding of TPM cycles defined in TPM1.2 spec. Note that this bit and [TpmLegacy] are independent bits; they respectively turn on decoding of different TPM addresses.										

#### D14F3x84 TMKBC\_BaseAddrLow

Bits	Description
31:7	<b>TmkbcBaseAddrLow.</b> Read-write. Reset: 0. This register defines the lower 32 bit memory address used for the TMKBC function. There are actually four sets of such mapping. The selection is controlled by <a href="#">D14F3x7C</a> [TmkbcSel].
6	<b>MaskBits13thru8.</b> Read-write. Reset: 0. Defines whether TMKBC address bits [13:8] are masked as do-not-care bits. 1=Masked. 0=No mask.
5	<b>MaskBits12thru8.</b> Read-write. Reset: 0. Defines whether TMKBC address bits [12:8] are masked as do-not-care bits. 1=Masked. 0=No mask.
4	<b>MaskBits11thru8.</b> Read-write. Reset: 0. Defines whether TMKBC address bits [11:8] are masked as do-not-care bits. 1=Masked. 0=No mask.
3	<b>MaskBits10thru8.</b> Read-write. Reset: 0. Defines whether TMKBC address bits [10:8] are masked as do-not-care bits. 1=Masked. 0=No mask.
2	<b>Addr64.</b> Read-write. Reset: 0. Defines whether the TMKBC address is 32 or 64 bits. 1=The address is 64-bit. 0=The address is 32-bit.
1:0	Reserved.

**D14F3x88 TMKBC\_BaseAddrHigh**

Bits	Description
31:0	<b>TmkbcBaseAddrHigh</b> . Read-write. Reset: 0. If <a href="#">D14F3x84[Addr64]</a> == 1, this register defines the upper 32 bit memory address used for the TMKBC function. If <a href="#">D14F3x84[Addr64]</a> == 0, this register has no meaning. There are actually four sets of such mapping. The selection is controlled by <a href="#">D14F3x7C[TmkbcSel]</a> .

**D14F3x8C TMKBC\_Remap**

Bits	Description
31:16	Reserved.
15:8	<b>TmkbcRemap</b> . Read-write. Reset: 0. This register defines the remap address [15:8] on the LPC bus. There are actually four sets of such mapping. The selection is controlled by <a href="#">D14F3x7C[TmkbcSel]</a> .
7:0	Reserved.

**D14F3x90 Wide IO 2**

Bits	Description
31:16	Reserved.
15:0	<b>IOBaseAddress2: IO Base Address 2</b> . Read-write. Reset: 0. 16-bit PCI IO base address for wide generic IO port range. This function is enabled by <a href="#">D14F3x48[WideIO2Enable]</a> . If <a href="#">D14F3x74[AlternativeWideIO2RangeEnable]</a> == 1, the range is 16 bytes; else, the ranges is 512 bytes.

**D14F3x98 EC\_LPC\_Cntrl**

Bits	Description
31:9	Reserved.
8	<b>EcHoldLpc</b> . Read-write. Reset: 0. 1=IMC holds the LPC bridge and host cannot access LPC.
7:1	Reserved.
0	<b>HostHoldLpc</b> . Read-write. Reset: 0. 1=Host holds the LPC bridge and prevents IMC from accessing LPC.

**D14F3xA0 SPI Base\_Addr**

Bits	Description
31:6	<b>SpiBaseAddr[31:6]</b> . Read-write. Reset: 0. BIOS: See 2.15.4. This register specifies the MMIO base address for the SPI ROM controller registers. See 3.25.9.2 [SPI Registers].
5:4	Reserved.
3	<b>RouteTpm2Spi</b> . Read-write. Reset: 0. 1=TPM cycles are routed to SPI bus with TPM_SPI_CS# asserted.

2	Reserved.
1	<b>SpiRomEnable.</b> Read-write. Reset: 1. 1=SPI ROM is enabled if chip is strapped to SPI ROM. 0=SPI ROM is disabled.
0	Reserved.

**D14F3xA4 EC\_PortAddress**

Bits	Description
31:16	Reserved.
15:1	<b>EcPortAddr15_1.</b> IF (D14F3xB8[EcPortHostAccessEn]==0) THEN Read-only. ELSE Read-write. ENDIF. Reset: 17h. If this field is non-zero, [EcPortActive] is set to 1, and an IO cycle from host has address[15:1] == EcPortAddr15_1, the cycle is routed to IMC.
0	<b>EcPortActive.</b> Read-write. Reset: 1. 1=LPC can decode the address specified in [EcPortAddr15_1]. 0= LPC ignores it.

**D14F3xB0 RomDmaSrcAddr**

Bits	Description
31:6	<b>DmaStartAddr.</b> Read-write. Reset: 0. Specifies the starting DMA address to read from the ROM. Note this is not the same as the legacy DMA function. This is meant to be used by BIOS to fetch BOOT code quicker.
5:0	Reserved.

**D14F3xB4 RomDmaDstAddr**

Bits	Description
31:6	<b>DmaDstAddr.</b> Read-write. Reset: 0. Specifies the target DMA address to be written in the system memory. Note this is not the same as the legacy DMA function. This is meant to be used by BIOS to fetch BOOT code quicker.
5:0	Reserved.

**D14F3xB8 RomDmaControl/EcControl/HostControl**

Bits	Description
31:30	Reserved.
29	<b>PrefetchMissEnHost.</b> Read-write. Reset: 1. 1=Force the on-going prefetch to stop if the address of the pending read is not within the prefetch range for host.
28	<b>PrefetchArbNoSwitch.</b> Read-write. Reset: 1. 1=Only allow prefetch.
27	<b>T_start_fix.</b> Read-write. Reset: 0. BIOS: 1. 1=Enable hand-instance of the pulse generator; allow LPCCLK0/LPCCLK1 to stop under ClkRun# protocol.

26	<b>DisableSpiInSleep.</b> Read-write. Reset: 0. BIOS: 1. 1=SPI signals are driven to low when IMC is not enabled in S3 or S5 and PwrGood goes to low. 0=SPI signals are left high when IMC is not enabled in S3 or S5 and PwrGood goes to low.
25	<b>LpcBusPullUpEn.</b> Read-write. Reset: 0. BIOS: 1. 0=Disable LAD[3:0] pull-up. 1=Enable LAD Pull-up when system is in S0 or when system is in S3/S5 and strapped to LPC ROM and LPC ROM is needed in S3/S5.
24	<b>PrefetchEnSpiFromHost.</b> Read-write. Reset: 0. BIOS: 1. This is for performance enhancement purpose. 1=SPI controller prefetches from the flash on behalf of the host.
23	<b>PrefetchEnSpiFromUSB.</b> Read-write. Reset: 0. BIOS: 1. This is for performance enhancement purpose. 1=SPI controller prefetches from the flash on behalf of USB3 controller.
22	<b>PrefetchEnSel.</b> Read-write. Reset: 1. 0=Prefetch only the address of the read request that is on the 4 byte boundary. 1=Prefetch only the address of the read request that is on the 8 byte boundary.
21	<b>PrefetchMissEn.</b> Read-write. Reset: 1. 1=Force the on-going prefetch to stop if the address of the pending read is not within the prefetch range for IMC.
20	<b>EcReadOfSwitch.</b> Read-write. Reset: 0. 1=Upper 16 bits of IMC read request address are specified in <a href="#">D14F3xC4 [EcRomRdOffset]</a> . 0=Upper 16 bits of IMC read request address are specified by auto-rom detection logic, see <a href="#">D14F3xCC [AutoRomCfg]</a> for auto-rom detection.
19	<b>EcPortHostAccessEn.</b> Read-write. Reset: 0. 1=Allow host to program <a href="#">D14F3xA4 [EC_PortAddress]</a> .
18	<b>PrefetchEnSpiFromEC.</b> Read-write. Reset: 0. 1=Enable prefetching a cache line (64 bytes) when IMC reads code from the SPI ROM. This bit can be programmed by IMC only.
17	<b>SpiHoldOnGevent9Dis.</b> Read-write. Reset: 0. 0=GEVENT9 and ROM_RST# are configured to SPI_HOLD# and SPI_WP# function. 1=GEVENT9 and ROM_RST# are not configured to SPI_HOLD# and SPI_WP# function.
16	<b>ECPageProtect.</b> Read-write. Reset: 0. Enable protection of the IMC page registers. 1= <a href="#">D14F3x[C7:BC]</a> are only read/writable by IMC; host writes have no effect and host reads return 0. 0= <a href="#">D14F3x[C7:BC]</a> are read/writable by both IMC and host.
15:6	<b>DWCCount.</b> Read-write. Reset: 0. This register specifies the number of cache lines (64 bytes) to be fetched from the ROM when DMA is used.
5:3	Reserved.
2	<b>RomcpSupportAbRetry.</b> Read-write. Reset: 1. 0=ROM copy doesn't support AB retry. 1=ROM copy supports AB retry.
1	<b>DmaErrorStatus.</b> Read; write-1-to-clear. Reset: 0. 1=Previous transfer has error. 0=Previous transfer has completed successfully.
0	<b>DmaStart.</b> Read; write-1-only; cleared-by-hardware. Reset: 0. Writing 1 to this bit causes LPC bridge to start the DMA function, with starting addresses defined by <a href="#">D14F3xB0</a> and <a href="#">D14F3xB4</a> . This bit returns the status of the DMA transfer. A return value of 0 means the DMA transfer is complete. A return value of 1 means the DMA transfer is running.

### D14F3xC0 EcRomWrOffset

Bits	Description
31:0	<b>EcRomWrOffset.</b> Read-write. Reset: FFF20000h. Specifies the address of IMC write requests. If <a href="#">D14F3xB8[ECPageProtect]</a> == 1, this register returns its current value. If <a href="#">D14F3xB8[ECPageProtect]</a> == 0, this register returns all 0s.

**D14F3xC4 EcRomRdOffset**

Bits	Description
31:0	<b>EcRomRdOffset.</b> Read-write. Reset: FFF20000h. Specifies the address of IMC read requests. If <b>D14F3xB8[ECPageProtect]</b> == 1, this register returns its current value. If <b>D14F3xB8[ECPageProtect]</b> == 0, this register returns all 0s.

**D14F3xC8 ClientRomProtect**

Bits	Description
31	<b>AutoSizeStart.</b> Read; write-1-only. Reset: 0. IF (READ) THEN This bit returns the status of AutoRom detection. 1=The AutoRom detection is done. 0=AutoRom detection is not done yet. ELSE 1=Trigger AutoRom detection. 0=No effect. ENDIF.
30:8	Reserved.
7:3	Reserved.
2	<b>UsbRomProtectEn.</b> Read-write. Reset: 0. 1=Software cannot access USB portion of the flash.
1	Reserved.
0	<b>EcRomPortectEn.</b> Read-write. Reset: 0. 1=Software cannot access IMC portion of the flash.

**D14F3xCC AutoRomCfg**

Bits	Description										
31:2	<b>AutoRomAddr.</b> Read-only. Reset: X. Based on the setting of AutoAddressSelect, this register returns the address value detected by the auto-rom detection logic (EcCodeOffset, XhciRomOffset).										
1:0	<b>AutoAddressSelect.</b> Read-write. Reset: 0. The FCH auto detects the IMC and USB3 firmware locations. After detection, it writes the address pointers of each firmware onto bits[31:2]. This field configures which address to be returned on bits[31:2]. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Reserved</td> </tr> <tr> <td>01b</td> <td>IMC</td> </tr> <tr> <td>10b</td> <td>Reserved</td> </tr> <tr> <td>11b</td> <td>USB XHCI</td> </tr> </tbody> </table>	Bits	Definition	00b	Reserved	01b	IMC	10b	Reserved	11b	USB XHCI
Bits	Definition										
00b	Reserved										
01b	IMC										
10b	Reserved										
11b	USB XHCI										

**D14F3xD0 ClkCntrl**

Bits	Description
31	<b>ClkRunEn.</b> Read-write. Reset: 0. BIOS: 1. 1=ClkRun function is enabled and LPCCLK0/LPCCLK1 can be stopped. 0=ClkRun function is disabled and LPCCLK0/LPCCLK1 can be running all the time.
30:24	<b>ClkRunDlyCounter.</b> Read-write. Reset: 8h. Specifies the amount of clocks to be extended before stopping the LPCCLK0/LPCCLK1.
23	Reserved. Reset: 1.
22	<b>Lclk1ClkrunOvrid.</b> Read-write. Reset: 1. 0=LPCCLK1 will be functioning with CLKRUN protocol. 1=LPCCLK1 will be forced running.

21	<b>Lclk0ClkrunOvrid.</b> Read-write. Reset: 1. 0=LPCCLK0 will be functioning with CLKRUN protocol. 1=LPCCLK0 will be forced running.										
20:15	Reserved. Reset: 3Fh.										
14	<b>Lclk1En.</b> Read-write. Reset: 1. 0=LPCCLK1 will be forced to stop. 1=LPCCLK1 will be functioning with CLKRUN protocol.										
13	<b>Lclk0En.</b> Read-write. Reset: 1. 0=LPCCLK0 will be forced to stop. 1=LPCCLK0 will be functioning with CLKRUN protocol.										
12:3	Reserved. Reset: 3E0h.										
2	<b>ClkRunDisable.</b> Read-write. Reset: 0. 1=CLKRUN# assertion is not allowed; In other words, the internal CLKRUN# coming from LPC is always 0, which means ClkRun logic can never stop the clock.										
1:0	<b>ClkGateCntrl.</b> Read-write. Reset: 2h. These two bits control whether the LPC module allows clock gating to the internal 66MHz core clock. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Disable the clock gating function</td> </tr> <tr> <td>01b</td> <td>Wait 16 clocks before allowing clock gating to the LPC module</td> </tr> <tr> <td>10b</td> <td>Wait 64 clocks before allowing clock gating to the LPC module</td> </tr> <tr> <td>11b</td> <td>Wait 256 clocks before allowing clock gating to the LPC module</td> </tr> </tbody> </table>	Bits	Definition	00b	Disable the clock gating function	01b	Wait 16 clocks before allowing clock gating to the LPC module	10b	Wait 64 clocks before allowing clock gating to the LPC module	11b	Wait 256 clocks before allowing clock gating to the LPC module
Bits	Definition										
00b	Disable the clock gating function										
01b	Wait 16 clocks before allowing clock gating to the LPC module										
10b	Wait 64 clocks before allowing clock gating to the LPC module										
11b	Wait 256 clocks before allowing clock gating to the LPC module										

### D14F3xD4 ClkRunOption

Bits	Description
31:8	Reserved.
7:4	<b>MinAssertion.</b> Read-write. Reset: 4h. Specifies the minimum time of ClkRun# assertion. The unit is 30 ns.
3:1	Reserved.
0	<b>ClkRunInputDly.</b> Read-write. Reset: 0. 0=Delay 30 ns before capturing ClkRun# input. 1=Delay 60 ns before capturing ClkRun# input.

### 3.25.9.2 SPI Registers

SPI configuration registers are accessed through SPI base address specified by [D14F3xA0 \[SPI Base\\_Addr\]](#). Software can communicate with the SPI ROM through the default memory or alternate program method:

- Memory access to the BIOS ROM address space is automatically handled by the hardware. The SPI ROM controller translates the memory address onto the SPI bus and accesses the SPI ROM data. Any other commands besides memory read or memory write to the SPI ROM need to go through the alternate program method.
- In alternate program method, software needs to program the SpiOpCode, SpiAddress, TxByteCount, RxByteCount, put the data into the transmit FIFO, and then execute the command. The hardware communicates with the SPI ROM using these parameters. This alternate method basically allows software to issue any flash vendor specific commands such as ERASE and STATUS.

### SPIx00 SPI\_Cntrl0

Bits	Description
31	<b>SpiBusy.</b> Read-only; updated-by-hardware. Reset: 0. 0=SPI bus is idle. 1=SPI bus is busy.



30:29	<b>SpiReadMode[2:1]</b> . Read-write. Reset: 0. This field, along with SpiReadMode[0] below, combine to specify the SPI read mode: <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>Normal read (up to 33M)</td> </tr> <tr> <td>001b</td> <td>Reserved</td> </tr> <tr> <td>010b</td> <td>Dual IO (1-1-2)</td> </tr> <tr> <td>011b</td> <td>Quad IO (1-1-4)</td> </tr> <tr> <td>100b</td> <td>Dual IO (1-2-2)</td> </tr> <tr> <td>101b</td> <td>Quad IO (1-4-4)</td> </tr> <tr> <td>110b</td> <td>Normal read (up to 66M)</td> </tr> <tr> <td>111b</td> <td>Fast Read</td> </tr> </tbody> </table>	Bits	Definition	000b	Normal read (up to 33M)	001b	Reserved	010b	Dual IO (1-1-2)	011b	Quad IO (1-1-4)	100b	Dual IO (1-2-2)	101b	Quad IO (1-4-4)	110b	Normal read (up to 66M)	111b	Fast Read
Bits	Definition																		
000b	Normal read (up to 33M)																		
001b	Reserved																		
010b	Dual IO (1-1-2)																		
011b	Quad IO (1-1-4)																		
100b	Dual IO (1-2-2)																		
101b	Quad IO (1-4-4)																		
110b	Normal read (up to 66M)																		
111b	Fast Read																		
28	<b>SpiClkGate</b> . Read-write. Reset: 0. 1=Skip the 8th SPI clock at the end data when doing read.																		
27	Reserved.																		
26:24	<b>ArbWaitCount</b> . Read-write. Reset: 7h. Specifies the amount of wait time the SPI controller asserts HOLD# before it should access the SPI ROM, under ROM sharing mode with the MAC.																		
23	<b>SpiHostAccessRomEn</b> . Read-write. Reset: 1. This is a clear-once protection bit; once it is cleared to 0 it cannot be set back to 1. 0=MAC cannot access BIOS ROM space (upper 512KB). 1=MAC can access BIOS ROM space.																		
22	<b>SpiAccessMacRomEn</b> . Read-write. Reset: 1. This is a clear-once protection bit. 0=Software cannot access MAC's portion of the ROM space (lower 512KB). 1=Software can access MAC's portion of the ROM space.																		
21	<b>IllegalAccess</b> . Read-only; updated-by-hardware. Reset: 0. 0=Legal index mode access. 1=Illegal index mode access.																		
20	<b>FifoPtrClr</b> . RAZ; write-1-only. Reset: 0. Writing 1 to this bit clears the internal FIFO pointer at SPIx0C[FifoPtr].																		
19	<b>SpiArbEnable</b> . Read-write. Reset: 0. If a MAC is sharing the ROM with the FCH, both chips need to go through an arbitration process before either one can access the ROM. If MAC is not sharing the SPI ROM, BIOS should set this bit to 0 to speed up the SPI ROM access. 1=Enable the arbitration.																		
18	<b>SpiReadMode[0]</b> . Read-write. Reset: 0. Bit 0 of SpiReadMode. See the definition of SpiReadMode[2:1] in this register.																		
17	Reserved.																		
16	<b>ExecuteOpCode</b> . Write-1-only; cleared-by-hardware. Reset: 0. Write 1 to execute the transaction in the alternate program registers. This bit returns to 0 when the transaction is complete. If the command is an illegal command, the bit cannot be set and thereby cannot execute.																		
15:8	Reserved.																		
7:0	<b>SpiOpCode</b> . Read-write. Reset: 0. Specifies the SPI opcode in alternate program method.																		

### SPIx04 SPI\_RestrictedCmd

Bits	Description
31:24	<b>RestrictedCmd3</b> . IF (SPIx00[SpiAccessMacRomEn]==0    SPIx00[SpiHostAccessRomEn]==0) THEN Read-only. ELSE Read-write. ENDIF. Reset: 0. Same as RestrictedCmd0.
23:16	<b>RestrictedCmd2</b> . IF (SPIx00[SpiAccessMacRomEn]==0    SPIx00[SpiHostAccessRomEn]==0) THEN Read-only. ELSE Read-write. ENDIF. Reset: 0. Same as RestrictedCmd0.

15:8	<b>RestrictedCmd1.</b> IF ( $SPIx00[SpiAccessMacRomEn]==0 \parallel SPIx00[SpiHostAccessRomEn]==0$ ) THEN Read-only. ELSE Read-write. ENDIF. Reset: 0. Same as RestrictedCmd0.
7:0	<b>RestrictedCmd0.</b> IF ( $SPIx00[SpiAccessMacRomEn]==0 \parallel SPIx00[SpiHostAccessRomEn]==0$ ) THEN Read-only. ELSE Read-write. ENDIF. Reset: 0. This specifies a restricted command issued by the MAC which is checked by the FCH. If the opcode issued by the MAC matches with this field and the address space is in the BIOS space, the SPI controller jams the entire interface as an attempt to stop that transaction.

**SPIx08 SPI\_RestrictedCmd2**

Bits	Description
31:24	<b>RestrictedCmdWoAddr2.</b> IF ( $SPIx00[SpiAccessMacRomEn]==0 \parallel SPIx00[SpiHostAccessRomEn]==0$ ) THEN Read-only. ELSE Read-write. ENDIF. Reset: 0. Same as [RestrictedCmdWoAddr0].
23:16	<b>RestrictedCmdWoAddr1.</b> IF ( $SPIx00[SpiAccessMacRomEn]==0 \parallel SPIx00[SpiHostAccessRomEn]==0$ ) THEN Read-only. ELSE Read-write. ENDIF. Reset: 0. Same as [RestrictedCmdWoAddr0].
15:8	<b>RestrictedCmdWoAddr0.</b> IF ( $SPIx00[SpiAccessMacRomEn]==0 \parallel SPIx00[SpiHostAccessRomEn]==0$ ) THEN Read-only. ELSE Read-write. ENDIF. Reset: 0. Same as $SPIx04[RestrictedCmd0]$ except that this field defines a restricted command that does not have address.
7:0	<b>RestrictedCmd4.</b> IF ( $SPIx00[SpiAccessMacRomEn]==0 \parallel SPIx00[SpiHostAccessRomEn]==0$ ) THEN Read-only. ELSE Read-write. ENDIF. Reset: 0. Same as $SPIx04[RestrictedCmd0]$ .

**SPIx0C SPI\_Cntrl1**

Bits	Description
31:24	<b>ByteCommand.</b> Read-write. Reset: 0. Specifies the command byte for the opcode transaction.
23:22	Reserved.
21:16	<b>WaitCount.</b> Read-write. Reset: 0. Specifies the time unit =15 ns * (WaitCount+1).
15:12	Reserved.
11	<b>TrackMacLockEn.</b> Read-write. Reset: 0. 1=The SPI controller locks the SPI from the MAC when it has detected a command from the MAC matching the value defined in $SPIx10[MacLockCmd0]$ or $SPIx10[MacLockCmd1]$ . 0=The SPI controller unlocks the bus when it has detected a command from the MAC matching the value defined in $SPIx10[MacUnlockCmd0]$ or $SPIx10[MacUnlockCmd1]$ .
10:8	<b>FifoPtr.</b> Read-only. Reset: 0. This specifies the internal pointer location, which can be cleared through $SPIx00[FifoPtrClr]$ .
7:0	<b>SpiParameters.</b> Read-write. Reset: 0. This is the TX/RX FIFO port which can take up to 8 bytes. To send data to SPI ROM, software writes data into this port. To retrieve data that are received from the SPI ROM, software reads from this port.

**SPIx10 SPI\_CmdValue0**

Bits	Description
31:24	<b>MacUnlockCmd1</b> . IF ( <b>SPIx00</b> [SpiAccessMacRomEn]==0    <b>SPIx00</b> [SpiHostAccessRomEn]==0) THEN Read-only. ELSE Read-write. ENDIF. Reset: 4h. Same as MacUnlockCmd0
23:16	<b>MacUnlockCmd0</b> . IF ( <b>SPIx00</b> [SpiAccessMacRomEn]==0    <b>SPIx00</b> [SpiHostAccessRomEn]==0) THEN Read-only. ELSE Read-write. ENDIF. Reset: 4h. This field is used to compare against the opcode sent out by the MAC. If <b>SPIx0C</b> [TrackMacLockEn]==1, the controller unlocks the SPI bus for the MAC. In other words, access by the CPU is allowed again.
15:8	<b>MacLockCmd1</b> . IF ( <b>SPIx00</b> [SpiAccessMacRomEn]==0    <b>SPIx00</b> [SpiHostAccessRomEn]==0) THEN Read-only. ELSE Read-write. ENDIF. Reset: 20h. Same as MacLockCmd0.
7:0	<b>MacLockCmd0</b> . IF ( <b>SPIx00</b> [SpiAccessMacRomEn]==0    <b>SPIx00</b> [SpiHostAccessRomEn]==0) THEN Read-only. ELSE Read-write. ENDIF. Reset: 6h. This is used to compare against the opcode sent out by the MAC. If <b>SPIx0C</b> [TrackMacLockEn]==1, the controller locks the SPI bus for the MAC. In other words, the MAC has the exclusive access to the ROM and access by the CPU is delayed until this is unlocked. This allows the MAC to do a certain sequence of operations without interruption.

**SPIx14 SPI\_CmdValue1**

Bits	Description
31:24	<b>RDSR</b> . IF ( <b>SPIx00</b> [SpiAccessMacRomEn]==0    <b>SPIx00</b> [SpiHostAccessRomEn]==0) THEN Read-only. ELSE Read-write. ENDIF. Reset: 5h. This is used to compare against the opcode sent out by the MAC. This is a predefined value to decode for the RDSR (read status register) command from the MAC.
23:16	<b>RDID</b> . IF ( <b>SPIx00</b> [SpiAccessMacRomEn]==0    <b>SPIx00</b> [SpiHostAccessRomEn]==0) THEN Read-only. ELSE Read-write. ENDIF. Reset: 9Fh. This is used to compare against the opcode sent out by the MAC. This is a predefined value to decode for the RDID (read ID) command from the MAC.
15:8	<b>WRDI</b> . IF ( <b>SPIx00</b> [SpiAccessMacRomEn]==0    <b>SPIx00</b> [SpiHostAccessRomEn]==0) THEN Read-only. ELSE Read-write. ENDIF. Reset: 4h. This is used to compare against the opcode sent out by the MAC. This is a predefined value to decode for the WRDI (write disable) command from the MAC.
7:0	<b>WREN</b> . IF ( <b>SPIx00</b> [SpiAccessMacRomEn]==0    <b>SPIx00</b> [SpiHostAccessRomEn]==0) THEN Read-only. ELSE Read-write. ENDIF. Reset: 6h. This is used to compare against the opcode sent out by the MAC. This is a predefined value to decode for the WREN (write enable) command from the MAC.

**SPIx18 SPI\_CmdValue2**

Bits	Description
31:24	<b>BYTEWR</b> . IF ( <b>SPIx00</b> [SpiAccessMacRomEn]==0    <b>SPIx00</b> [SpiHostAccessRomEn]==0) THEN Read-only. ELSE Read-write. ENDIF. Reset: 2h. This is used to compare against the opcode sent out by the MAC. This is a predefined value to decode for the BYTEWR (byte write) command from the MAC.
23:16	Reserved.

15:8	<b>FRead.</b> IF (SPIx00[SpiAccessMacRomEn]==0    SPIx00[SpiHostAccessRomEn]==0) THEN Read-only. ELSE Read-write. ENDIF. Reset: Bh. This is used to compare against the opcode sent out by the MAC. This is a predefined value to decode for the FRead (fast read) command from the MAC.
7:0	<b>Read.</b> IF (SPIx00[SpiAccessMacRomEn]==0    SPIx00[SpiHostAccessRomEn]==0) THEN Read-only. ELSE Read-write. ENDIF. Reset: 3h. This is used to compare against the opcode sent out by the MAC. This is a predefined value to decode for the Read (read byte) command from the MAC.

**SPIx1C Reserved**

Bits	Description
7:0	Reserved. Reset: FFh.

**SPIx1D Alt\_SPI\_CS**

Bits	Description										
7	<b>SpiCsDlySel.</b> Read-write. Reset: 0. 1=125 ns minimum SPI_CS# de-assertion time. 0=75 ns minimum SPI_CS# de-assertion time.										
6	Reserved.										
5	<b>SpiProtectLock.</b> Read-write. Reset: 0. 1=bit 3, 4, and 5 are no longer writable.										
4	<b>SpiProtectEn1.</b> IF (SPIx1D[SpiProtectLock]==1) THEN Read-only. ELSE Read-write. ENDIF. Reset: 0. 1=Enable SPI protection to prevent host from accessing IMC and USB3 space.										
3	<b>SpiProtectEn0.</b> IF (SPIx1D[SpiProtectLock]==1) THEN Read-only. ELSE Read-write. ENDIF. Reset: 0. 1=Enable SPI read/write protection ranges specified by D14F3x[5C,58,54,50].										
2	<b>WriteBufferEn.</b> Read-write. Reset: 0. SPI write performance enhancement. 1=SPI bridge can take burst write from the host and transfer it to the SPI flash.										
1:0	<b>AltSpiCsEn.</b> Read-write. Reset: 0. These two bits enable the alternate SPI_CS#. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>select xSPI_CS#</td> </tr> <tr> <td>01b</td> <td>select xSPI_CS1#</td> </tr> <tr> <td>10b</td> <td>select xSPI_CS2#</td> </tr> <tr> <td>11b</td> <td>select xSPI_CS3#</td> </tr> </tbody> </table>	Bits	Definition	00b	select xSPI_CS#	01b	select xSPI_CS1#	10b	select xSPI_CS2#	11b	select xSPI_CS3#
Bits	Definition										
00b	select xSPI_CS#										
01b	select xSPI_CS1#										
10b	select xSPI_CS2#										
11b	select xSPI_CS3#										

**SPIx1E SpiExtRegIndx**

Bits	Description
7:0	<b>SPI_ExtReg_Indx.</b> Read-write. Reset: 0. Specifies the offset of the extended SPI register to be read/written from SPIx1F [SpiExtRegData].

**SPIx1F SpiExtRegData**

Bits	Description
7:0	<b>SPI_ExtReg_Data.</b> Read-write. Reset: 00h. Specifies the read data or write data of the extended SPI register.

**SPIx1F\_x00 DDR\_CMD**

Bits	Description
7:0	<b>DDR_CMD</b> . Read-write. Reset: 3Bh. Specifies the command value to be used when host is doing a Dual Output Read. This is programmable to accommodate different opcodes.

**SPIx1F\_x01 QDR\_CMD**

Bits	Description
7:0	<b>QDR_CMD</b> . Read-write. Reset: 6Bh. Specifies the command value to be used when host is doing a Quad Output Read. This is programmable to accommodate different opcodes.

**SPIx1F\_x02 DPR\_CMD**

Bits	Description
7:0	<b>DPR_CMD</b> . Read-write. Reset: BBh. Specifies the command value to be used when host is doing a Dual IO High Performance Read. This is programmable to accommodate different opcodes.

**SPIx1F\_x03 QPR\_CMD**

Bits	Description
7:0	<b>QPR_CMD</b> . Read-write. Reset: EBh. Specifies the command value to be used when host is doing a Quad IO High Performance Read. This is programmable to accommodate different opcodes.

**SPIx1F\_x04 ModeByte**

Bits	Description
7:0	<b>ModeByte</b> . Read-write. Reset:0. Bits [7:4] of this field control the length of the next Dual/Quad IO High Performance instruction through the inclusion or exclusion of the first byte instruction code. Bits [3:0] of this field are don't care bits ("x"). If ModeByte == Axh, then the device remains in Dual IO High Performance Read Mode and the next address can be entered (after CS# is raised high and then asserted low) without requiring the instruction opcode. Whenever DPR_CMD or QPR_CMD is used, ModeByte is also sent out onto the SPI stream.

**SPIx1F\_x05 TxByteCount**

Bits	Description
7:0	<b>TxByteCount</b> . Read-write. Reset: 0. Specifies the number of bytes to be sent to SPI ROM.

**SPIx1F\_x06 RxByteCount**

Bits	Description
7:0	<b>RxByteCount.</b> Read-write. Reset: 0. Specifies the number of bytes to be received from the SPI ROM.

**SPIx1F\_x07 SPIDataFifoPtr**

Bits	Description
7:0	<b>SpiDataFiFoPtr.</b> Read-only. Reset: 0. Specifies the current pointer of read/write Data FIFO.

**SPIx20 SPI100 Enable**

Bits	Description
7:1	Reserved.
0	<b>UseSpi100.</b> Read-write. Reset: 0. BIOS: see 2.15.4.2. 0=Do not support 100 MHz speed. 1=Support 100 MHz speed. The actual read speed also depends on SPIx22.

**SPIx22 SPI100 Speed Config**

Bits	Description																				
15:12	<p><b>NormSpeed[3:0].</b> Read-write. Reset: 0011b . Configures the SPI bus normal speed in SPI100 engine. If the command is not using TpmSpeed and FastSpeed, it will use NormSpeed.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>66.66 MHz</td> <td>100b</td> <td>100 MHz</td> </tr> <tr> <td>001b</td> <td>33.33 MHz</td> <td>101b</td> <td>Reserved</td> </tr> <tr> <td>010b</td> <td>22.22 MHz</td> <td>110b</td> <td>Reserved</td> </tr> <tr> <td>011b</td> <td>16.66 MHz</td> <td>111b</td> <td>800 KHz</td> </tr> </tbody> </table>	Bits	Definition	Bits	Definition	000b	66.66 MHz	100b	100 MHz	001b	33.33 MHz	101b	Reserved	010b	22.22 MHz	110b	Reserved	011b	16.66 MHz	111b	800 KHz
Bits	Definition	Bits	Definition																		
000b	66.66 MHz	100b	100 MHz																		
001b	33.33 MHz	101b	Reserved																		
010b	22.22 MHz	110b	Reserved																		
011b	16.66 MHz	111b	800 KHz																		
11:8	<p><b>FastSpeed[3:0].</b> Read-write. Reset: 0001b. Configures the SPI bus speed for the following command in SPI100 engine:</p> <ul style="list-style-type: none"> <li>• FAST READ</li> <li>• DDR READ (1-1-2)</li> <li>• QDR READ (1-1-4)</li> <li>• DPR READ (1-2-2)</li> <li>• QPR READ (1-4-4)</li> </ul> <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>66.66 MHz</td> <td>100b</td> <td>100 MHz</td> </tr> <tr> <td>001b</td> <td>33.33 MHz</td> <td>101b</td> <td>Reserved</td> </tr> <tr> <td>010b</td> <td>22.22 MHz</td> <td>110b</td> <td>Reserved</td> </tr> <tr> <td>011b</td> <td>16.66 MHz</td> <td>111b</td> <td>800 KHz</td> </tr> </tbody> </table>	Bits	Definition	Bits	Definition	000b	66.66 MHz	100b	100 MHz	001b	33.33 MHz	101b	Reserved	010b	22.22 MHz	110b	Reserved	011b	16.66 MHz	111b	800 KHz
Bits	Definition	Bits	Definition																		
000b	66.66 MHz	100b	100 MHz																		
001b	33.33 MHz	101b	Reserved																		
010b	22.22 MHz	110b	Reserved																		
011b	16.66 MHz	111b	800 KHz																		

7:4	<b>AltSpeed[3:0]</b> . Read-write. Reset: 0001b. Configures the SPI bus speed for the AltOpCode mode in SPI100 engine.			
	<u>Bits</u>	<u>Definition</u>	<u>Bits</u>	<u>Definition</u>
	000b	66.66 MHz	100b	100 MHz
	001b	33.33 MHz	101b	Reserved
	010b	22.22 MHz	110b	Reserved
	011b	16.66 MHz	111b	800 KHz
3:0	<b>TpmSpeed[3:0]</b> . Read-write. Reset: 0011b. Configures the SPI bus speed for TPM read and write in SPI100 engine.			
	<u>Bits</u>	<u>Definition</u>	<u>Bits</u>	<u>Definition</u>
	000b	66.66 MHz	100b	100 MHz
	001b	33.33 MHz	101b	Reserved
	010b	22.22 MHz	110b	Reserved
	011b	16.66 MHz	111b	800 KHz

**SPIx40 DDRCmdCode**

Bits	Description
7:0	<b>DDR_CMD</b> . Read-write. Reset: 3Bh. This is the shadow register of <a href="#">SPIx1F_x00</a> .

**SPIx41 QDRCmdCode**

Bits	Description
7:0	<b>QDR_CMD</b> . Read-write. Reset: 6Bh. This is the shadow register of <a href="#">SPIx1F_x01</a> .

**SPIx42 DPRCmdCode**

Bits	Description
7:0	<b>DPR_CMD</b> . Read-write. Reset: BBh. This is the shadow register of <a href="#">SPIx1F_x02</a> .

**SPIx43 QPRCmdCode**

Bits	Description
7:0	<b>QPR_CMD</b> . Read-write. Reset: EBh. This is the shadow register of <a href="#">SPIx1F_x03</a> .

**SPIx44 ModeByte**

Bits	Description
7:0	<b>ModeByte</b> . Read-write. Reset: 0. This is the shadow register of <a href="#">SPIx1F_x04</a> .

**SPIx45 CmdCode**

Bits	Description
7:0	<b>SpiOpCode</b> . Read-write. Reset: 0. This is the shadow register of SPIx00[SpiOpCode].

**SPIx47 CmdTrigger**

Bits	Description
7	<b>Excute</b> . Write-1-only; cleared-by-hardware. Reset: 0. This bit is the shadow register of SPIx00[ExcuteOpCode].
6:0	Reserved.

**SPIx48 TxByteCount**

Bits	Description
7:0	<b>TxByteCount</b> . Read-write. Reset: 0. This is the shadow register of SPIx1F_x05.

**SPIx4B RxByteCount**

Bits	Description
7:0	<b>RxByteCount</b> . Read-write. Reset: 0. This is the shadow register of SPIx1F_x06.

**SPIx4C SpiStatus**

Bits	Description
31	<b>SpiBusy</b> . Read-only; updated-by-hardware. Reset: 0. 1=SPI bus is busy. 0=SPI bus is idle.
30:23	Reserved.
22:16	<b>FiFoRdPtr</b> . Read-only; updated-by-hardware. Reset: 0. The current Data FIFO read pointer.
15	Reserved.
14:8	<b>FiFoWrPtr</b> . Read-only; updated-by-hardware. Reset: 0. The current Data FIFO write pointer.
7:0	<b>DoneByteCount</b> . Read-only; updated-by-hardware. Reset: 0. Indicates how many bytes has been received or sent in the previous SPI transaction.

**SPIx[C6:80] FIFO[70:0]**

Bits	Description
7:0	<b>FiFo</b> . Read-write. Reset: 0. Contains the Data FiFo byte which is used in command mode to send or receive data.



**3.25.10 High Precision Event Timer (HPET) Registers**

HPET registers are accessed through two methods:

- Memory access through HPET base address defined by [PMx50 \[HPETEn\]](#):
  1. Program the HPET base address through [PMx50\[HpetBaseAddress\]](#).
  2. Enable HPET address decoding through [PMx50\[HpetDecodeEn\]](#).
- Memory mapped or IO mapped access through the AcpiMmio region. The HPET registers range from [PMx24\[AcpiMmioAddr\]+C00h](#) to [PMx24\[AcpiMmioAddr\]+CFFh](#). See [PMx24 \[AcpiMmioEn\]](#).

**HPETx000 ID**

Bits	Description
31:16	<b>VendorID</b> . Read-only. Reset: 1022h.
15	<b>LegacyCap</b> . Read-only. Reset: 1. Legacy replacement interrupt is supported.
14	Reserved.
13	<b>CounterSizeCap</b> . Read-only. Reset: 0. Main counter is 32-bits wide and cannot operate in 64-bit mode.
12:8	<b>NumTmrCap</b> . Read-only. Reset: 2h. Three timers are supported.
7:0	<b>RevID</b> . Read-only. Reset: 10h. Revision ID.

**HPETx004 ClkPeriod**

Bits	Description
31:0	<b>CounterClkPeriod</b> . Read-only; updated-by-hardware. Reset: 429B17Eh. Specifies the clock period of each HPET timer tick. HPET main counter runs at 14.31818 MHz. The unit is femtosecond ( $10^{-15}$ seconds). The value of this register can be modified through <a href="#">MISCx74</a> .

**HPETx010 Config**

Bits	Description
31:2	Reserved.
1	<b>LegacyEn</b> . Read-write. Reset: 0. 1=Timer0 interrupt goes to IRQ0 of PIC controller, INT2 of IOAPIC; Timer1 interrupt goes to IRQ8 of PIC controller, INT8 of IOAPIC.
0	<b>TmrEn</b> . Read-write. Reset: 0. 0=Pause main counter and disable all timer interrupts. 1=Allow main counter to run and allow timer interrupts if enabled.

**HPETx020 Interrupt Status**

Bits	Description
31:3	Reserved.
2	<b>Tmr2IntrSts</b> . IF (HPETx140[TmrIntTyp]==1) THEN Read; set-by-hardware; write-1-to-clear. ELSE Read-write. ENDIF. Reset: 0. 0=Timer2 interrupt is not active. 1=Timer2 interrupt is active. If Timer2 is set to edge-triggered mode, software should ignore this bit and always write 0 to this bit.

1	<b>Tmr1IntrSts.</b> IF (HPETx120[TmrIntTyp]==1) THEN Read; set-by-hardware; write-1-to-clear. ELSE Read-write. ENDIF. Reset: 0. 0=Timer1 interrupt is not active. 1=Timer1 interrupt is active. If Timer1 is set to edge-triggered mode, software should ignore this bit and always write 0 to this bit.
0	<b>Tmr0IntrSts.</b> IF (HPETx100[TmrIntTyp]==1) THEN Read; set-by-hardware; write-1-to-clear. ELSE Read-write. ENDIF. Reset: 0. 0=Timer0 interrupt is not active. 1=Timer0 interrupt is active. If Timer0 is set to edge-triggered mode, software should ignore this bit and always write 0 to this bit.

### HPETx0F0 Main Counter

Bits	Description
63:32	Reserved.
31:0	<b>MainCounter.</b> Read-write. Reset: 0. Specified the HPET main counter, incremented by 1 on every clock. Counter should be written to only when it is halted.

### HPETx1[4:0:Step2]0 Timer[2:0] Config Capability

Bits	Description
63:32	<b>TmrIntRouteCap.</b> Read-only. Reset: C00000h. Indicates which INT entry of IOAPIC can be assigned to the timer interrupt.
31:16	Reserved.
15	<b>TmrFsbCap.</b> Read-only. Reset: 1. 1=Front side bus delivery is supported.
14	<b>TmrFsbEn.</b> Read-write. Reset: 0. 1=Enable front side bus delivery of interrupt.
13:9	<b>TmrIntRoute.</b> Read-write. Reset: 0. This specifies which INT entry of IOAPIC the timer is routed to if <a href="#">HPETx010[LegacyEn]</a> == 0.
8	<b>Tmr32ModeEn.</b> Read-only. Reset: 0. 0=64-bit timer is not supported.
7	Reserved.
6	<b>TmrSetPer.</b> Read-write. Reset: 0. 1=Allow software to set the timer's accumulator if the timer is set to periodic mode. The bit is automatically cleared when <a href="#">HPETx1[4:0:Step2]8 [Timer[2:0] Comparator]</a> is written by software.
5	<b>TmrSizeCap.</b> Read-only. Reset: 0. The timer is 32-bits wide.
4	<b>TmrTypCap.</b> Read-only. Reset: 1. The timer supports periodic interrupt delivery mode.
3	<b>TmrTyp.</b> Read-write. Reset: 0. Selects the timer interrupt type. 0=Non-periodic. 1=Periodic
2	<b>TmrIntEn.</b> Read-write. Reset: 0. 1=Enable the timer interrupt.
1	<b>TmrIntTyp.</b> Read-write. Reset: 0. Specifies the timer interrupt polarity. 0=Edge triggered. 1=Level triggered
0	Reserved.

### HPETx1[4:0:Step2]8 Timer[2:0] Comparator

Bits	Description
------	-------------

63:32	Reserved.
31:0	<p><b>Comparator.</b> IF (HPETx1[4:0:Step2]0[TmrTyp] == 0) THEN Read-write. ELSEIF ((HPETx1[4:0:Step2]0[TmrTyp] == 1) &amp;&amp; (HPETx1[4:0:Step2]0[TmrSetPer] == 1)) THEN Read-write. ELSE Read-only. ENDIF. Reset: FFFF_FFFFh. Comparator is periodically incremented by the value last written to this register. This register is updated by hardware when (HPETx1[4:0:Step2]0[TmrTyp] == 1) &amp;&amp; the current value in this register matches with the value in HPETx0F0 [Main Counter].</p>

**3.25.11 Miscellaneous (MISC) Registers**

MISC register space is accessed through the AcpiMmio region, which can be memory mapped or IO mapped. The MISC registers range from **PMx24**[AcpiMmioAddr]+E00h to **PMx24**[AcpiMmioAddr]+EFFh. See **PMx24** [AcpiMmioEn].

**MISCx00 GPPClkCntrl**

BIOS: See 2.15.9.1.

Bits	Description																				
31	Reserved.																				
30:27	<b>CGIPLL_VREG_BIAS</b> . Read-write. Reset: 0. Voltage regulator current bias control (Spare).																				
26:25	<b>CGIPLL_VREG_CNTL</b> . Read-write. Reset: 0. Voltage regulator voltage control (Spare).																				
24	<b>CGIPLL_VREG_VREF_SEL</b> . Read-write. Reset: 0. Select Voltage regulator reference (Spare). 0=Bandgap. 1=Resistor Divider.																				
23	<b>CGIPLL_PLLCORE_PWDN_EN</b> . Read-write. Reset: 0. Control for powering down PLL's analog core (including PFD/CP, OTA, VCO, REFGNs, LF and all differential output drivers). It is only used for external clock-chip mode to bypass XTALIN 25 MHz to SATA differentially, when CG1_PLL is required to be off. 0=Disable. 1=Enable.																				
22:19	<b>CGPLL_BG_ADJ</b> . Read-write. Reset: 8. Fine adjust for band-gap. Level shifted inside PLL to 1.8V.																				
18:16	<b>CG_OSC_GAIN_EN</b> . Read-write. Reset: 6. CG_XTAL Pad Gain setting control of oscillator buffer.																				
15:12	<p><b>GPP_CLK3_ClockRequestMapping</b>. Read-write. Reset: Fh. GPP3 PCIe® clock pins (GPP_CLK3P/GPP_CLK3N) output control by CLK_REQ# pin. GPP_CLK3P/GPP_CLK3N pins are powered off when FCH is strapped to use an external clock, and powered on when FCH is strapped to operate in integrated clock mode. When FCH is in integrated clock mode, GPP3 PCIe® clock can be powered off according to the CLK_REQ mapping table below, and the selected CLK_REQ# input can power off the GPP3 PCIe® clock output pins if it is asserted.</p> <p>GPP3_CLKREQ_Mapping:</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>0000b</td> <td>Off</td> </tr> <tr> <td>0001b</td> <td>CLK_REQ0#</td> </tr> <tr> <td>0010b</td> <td>CLK_REQ1#</td> </tr> <tr> <td>0011b</td> <td>CLK_REQ2#</td> </tr> <tr> <td>0100b</td> <td>CLK_REQ3#</td> </tr> <tr> <td>1001b-1010b</td> <td>Reserved; Off</td> </tr> <tr> <td>1010b</td> <td>CLK_REQGfx#</td> </tr> <tr> <td>1110b-1011b</td> <td>Reserved; Off</td> </tr> <tr> <td>1111b</td> <td>On</td> </tr> </tbody> </table>	Bits	Definition	0000b	Off	0001b	CLK_REQ0#	0010b	CLK_REQ1#	0011b	CLK_REQ2#	0100b	CLK_REQ3#	1001b-1010b	Reserved; Off	1010b	CLK_REQGfx#	1110b-1011b	Reserved; Off	1111b	On
Bits	Definition																				
0000b	Off																				
0001b	CLK_REQ0#																				
0010b	CLK_REQ1#																				
0011b	CLK_REQ2#																				
0100b	CLK_REQ3#																				
1001b-1010b	Reserved; Off																				
1010b	CLK_REQGfx#																				
1110b-1011b	Reserved; Off																				
1111b	On																				

11:8	<p><b>GPP_CLK2_ClockRequestMapping.</b> Read-write. Reset: Fh. GPP2 PCIe® clock pins (GPP_CLK2P/GPP_CLK2N) output control by CLK_REQ# pin. GPP_CLK2P/GPP_CLK2N pins are powered off when FCH is strapped to use an external clock, and powered on when FCH is strapped to operate in integrated clock mode. When FCH is in integrated clock mode, GPP2 PCIe® clock can be powered off according to the CLK_REQ mapping table below, and the selected CLK_REQ# input can power off the GPP2 PCIe® clock output pins if it is asserted.</p> <p>GPP2_CLKREQ_Mapping:</p> <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>0000b</td> <td>Off</td> </tr> <tr> <td>0001b</td> <td>CLK_REQ0#</td> </tr> <tr> <td>0010b</td> <td>CLK_REQ1#</td> </tr> <tr> <td>0011b</td> <td>CLK_REQ2#</td> </tr> <tr> <td>0100b</td> <td>CLK_REQ3#</td> </tr> <tr> <td>1001b-1010b</td> <td>Reserved; Off</td> </tr> <tr> <td>1010b</td> <td>CLK_REQGfx#</td> </tr> <tr> <td>1110b-1011b</td> <td>Reserved; Off</td> </tr> <tr> <td>1111b</td> <td>On</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	0000b	Off	0001b	CLK_REQ0#	0010b	CLK_REQ1#	0011b	CLK_REQ2#	0100b	CLK_REQ3#	1001b-1010b	Reserved; Off	1010b	CLK_REQGfx#	1110b-1011b	Reserved; Off	1111b	On
<u>Bits</u>	<u>Definition</u>																				
0000b	Off																				
0001b	CLK_REQ0#																				
0010b	CLK_REQ1#																				
0011b	CLK_REQ2#																				
0100b	CLK_REQ3#																				
1001b-1010b	Reserved; Off																				
1010b	CLK_REQGfx#																				
1110b-1011b	Reserved; Off																				
1111b	On																				

7:4	<p><b>GPP_CLK1_ClockRequestMapping.</b> Read-write. Reset: Fh. GPP1 PCIe clock pins (GPP_CLK1P/GPP_CLK1N) output control by CLK_REQ# pin. GPP_CLK1P/GPP_CLK1N pins are powered off when FCH is strapped to use an external clock and powered on when FCH is strapped to operate in integrated clock mode. When FCH is in integrated clock mode, GPP1 PCIe clock can be powered off according to the CLK_REQ mapping table below, and the selected CLK_REQ# input can power off GPP1 PCIe clock output pins if it is asserted.</p> <p>GPP1_CLKREQ_Mapping:</p> <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr><td>0000b</td><td>Off</td></tr> <tr><td>0001b</td><td>CLK_REQ0#</td></tr> <tr><td>0010b</td><td>CLK_REQ1#</td></tr> <tr><td>0011b</td><td>CLK_REQ2#</td></tr> <tr><td>0100b</td><td>CLK_REQ3#</td></tr> <tr><td>1001b-0101b</td><td>Reserved; Off</td></tr> <tr><td>1010b</td><td>CLK_REQGfx#</td></tr> <tr><td>1110b-1011b</td><td>Reserved; Off</td></tr> <tr><td>1111b</td><td>On</td></tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	0000b	Off	0001b	CLK_REQ0#	0010b	CLK_REQ1#	0011b	CLK_REQ2#	0100b	CLK_REQ3#	1001b-0101b	Reserved; Off	1010b	CLK_REQGfx#	1110b-1011b	Reserved; Off	1111b	On
<u>Bits</u>	<u>Definition</u>																				
0000b	Off																				
0001b	CLK_REQ0#																				
0010b	CLK_REQ1#																				
0011b	CLK_REQ2#																				
0100b	CLK_REQ3#																				
1001b-0101b	Reserved; Off																				
1010b	CLK_REQGfx#																				
1110b-1011b	Reserved; Off																				
1111b	On																				
3:0	<p><b>GPP_CLK0_ClockRequestMapping.</b> Read-write. Reset: Fh. GPP0 PCIe clock pins (GPP_CLK0P/GPP_CLK0N) output control by CLK_REQ# pin. GPP_CLK0P/GPP_CLK0N pins are powered off when FCH is strapped to use an external clock and powered on when FCH is strapped to operate in integrated clock mode. When FCH is in integrated clock mode, GPP0 PCIe clock can be powered off according to the CLK_REQ mapping table below, and the selected CLK_REQ# input can power off the GPP0 PCIe clock output pins if it is asserted.</p> <p>GPP0_CLKREQ_Mapping:</p> <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr><td>0000b</td><td>Off</td></tr> <tr><td>0001b</td><td>CLK_REQ0#</td></tr> <tr><td>0010b</td><td>CLK_REQ1#</td></tr> <tr><td>0011b</td><td>CLK_REQ2#</td></tr> <tr><td>0100b</td><td>CLK_REQ3#</td></tr> <tr><td>1001b-0101b</td><td>Reserved; Off</td></tr> <tr><td>1010b</td><td>CLK_REQGfx#</td></tr> <tr><td>1110b-1011b</td><td>Reserved; Off</td></tr> <tr><td>1111b</td><td>On</td></tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	0000b	Off	0001b	CLK_REQ0#	0010b	CLK_REQ1#	0011b	CLK_REQ2#	0100b	CLK_REQ3#	1001b-0101b	Reserved; Off	1010b	CLK_REQGfx#	1110b-1011b	Reserved; Off	1111b	On
<u>Bits</u>	<u>Definition</u>																				
0000b	Off																				
0001b	CLK_REQ0#																				
0010b	CLK_REQ1#																				
0011b	CLK_REQ2#																				
0100b	CLK_REQ3#																				
1001b-0101b	Reserved; Off																				
1010b	CLK_REQGfx#																				
1110b-1011b	Reserved; Off																				
1111b	On																				

**MISCx04 ClkOutputCntrl**

Bits	Description
31	<b>CG2_RegulatorOff.</b> Read-write. Reset: 0. 1=Turn off CG2_PLL Regulator.
30	<b>CG1_RegulatorOff.</b> Read-write. Reset: 0. 1=Turn off CG1_PLL Regulator.
29	Reserved.
28	<b>PciePhyRefClkPwDn.</b> Read-write. Reset: 0. 1=Power-off Clkgen_outpad 100Mhz CML reference clock for PCIE_PHY.
27	<b>CLKGEN_TestClkPwDn.</b> Read-write. Reset: 0. 1=Power off Clkgen_outpad TestClk Output buffer.

26	<b>EXT_CLK_ClockSourceOverride.</b> Read-write. Reset: 0. EXT_CLK clock source override control. EXT_CLK clock source is controlled by strap (LPCCLK1). If Strap (LPCCLK1) == 1, clock source is from CG_PLL. If Strap (LPCCLK1) == 0, clock source is from external clock chip through GPP_CLK3_P/N pins. This override bit allows to invert the strap that controls EXT_CLK clock source. 0=Use the strap value (LPCCLK1) to determine whether EXT_CLK clock source from either CG_PLL or external clock chip. 1=Invert the strap that controls EXT_CLK clock source.										
25	<b>CG2PLL_CoreClkOutputEn.</b> Read-write. Reset: 1. CG2_PLL Core Clock (400 MHz) Output Enable. 0=Clock Output Buffer disable. 1=Clock Output Buffer Enable.										
24:23	<b>CG1PLL_DS_Order.</b> Read-write. Reset: 1. DS order setting. <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>disable</td> </tr> <tr> <td>01b</td> <td>1st order</td> </tr> <tr> <td>10b</td> <td>2nd order</td> </tr> <tr> <td>11b</td> <td>3rd order</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	00b	disable	01b	1st order	10b	2nd order	11b	3rd order
<u>Bits</u>	<u>Definition</u>										
00b	disable										
01b	1st order										
10b	2nd order										
11b	3rd order										
22	<b>CG1PLL_DS_PrbsEn.</b> Read-write. Reset: 1. 1=Enable CG1_PLL PRBS17 generator in DS modulator.										
21:20	<b>CG1PLL_FBDiv_FractionCtl.</b> Read-write. Reset: 1. CG1_PLL Slip Req/Ack handshaking control: <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Slip_req/Slip_ack handshake (8 VCO clock cycles per slip)</td> </tr> <tr> <td>01b</td> <td>Slip once every clock cycle (ignore slip_ack)</td> </tr> <tr> <td>10b</td> <td>Slip once every 2 clock cycles (ignore slip_ack)</td> </tr> <tr> <td>11b</td> <td>Slip once every 4 clock cycles (ignore slip_ack)</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	00b	Slip_req/Slip_ack handshake (8 VCO clock cycles per slip)	01b	Slip once every clock cycle (ignore slip_ack)	10b	Slip once every 2 clock cycles (ignore slip_ack)	11b	Slip once every 4 clock cycles (ignore slip_ack)
<u>Bits</u>	<u>Definition</u>										
00b	Slip_req/Slip_ack handshake (8 VCO clock cycles per slip)										
01b	Slip once every clock cycle (ignore slip_ack)										
10b	Slip once every 2 clock cycles (ignore slip_ack)										
11b	Slip once every 4 clock cycles (ignore slip_ack)										
19:17	Reserved.										
16	<b>ClockBufferBiasPowerDownEnable.</b> Read-write. Reset: 1. 1=Power down the clock buffers bias current circuit for power saving.										
15:14	Reserved.										
13	<b>PCIE_RCLK_PowerDownEnable.</b> Read-write. Reset: 0. BIOS: See 2.15.9.1. 1=Power down PCIE_RCLK input buffer for power saving.										
12	<b>GFX_GPP_CLK_ClockOutputOverride: GFX_GPP_CLK [2:0] ClockOutputOverride.</b> Read-write. Reset: 0. GFX_GPP_CLK[2:0] clock output override control. GFX_CLK and GPP_CLK[2:0] clock outputs are controlled by [SLT_GFX_CLKClockRequestMapping] and MISCx00[GPP_CLK0_ClockRequestMapping, GPP_CLK1_ClockRequestMapping, GPP_CLK2_ClockRequestMapping]. Basically it depends on whether chip is in internal (Strap (LPCCLK1) = 1) or external (Strap (LPCCLK1) = 0) clock mode. This override bit allows to invert the strap value that controls clock output buffer. 0=Use SLT_GFX Clock Request mapping and GPP_CLK[2:0] Clock Request mapping to determine whether clock output buffer is on or off. 1=Invert the strap that controls those clock output buffer.										
11	<b>GPP_CLK3_ClockOutputOverride.</b> Read-write. Reset: 0. GPP_CLK3 clock output override control. GPP_CLK3 is a bi-directional pin depending on LPCCLK1 strap value. If Strap (LPCCLK1) =1, GPP_CLK3 provides clock for external device and output buffer will be on. If Strap (LPCCLK1) =0, GPP_CLK3 receives clock from external clock chip and output buffer will be off. This override bit allows to invert the strap that controls GPP_CLK3 clock output buffer. 0=Use the strap value (LPCCLK1) to determine whether GPP_CLK3 clock output buffer is on or off. 1=Invert the strap that controls GPP_CLK3 clock output buffer.										

10	<b>CPU_CLK_ClockSourceOverride.</b> Read-write. Reset: 0. CPU_CLK clock source override control. CPU_CLK clock source is controlled by strap (LPCCLK1). If Strap (LPCCLK1) =1, CPU_CLK clock source is from CG_PLL. If Strap (LPCCLK1) =0, CPU_CLK clock source is from external clock chip through GPP_CLK3_P/N pins. This override bit allows to invert the strap that controls CPU_CLK clock source. 0=Use the strap value (LPCCLK1) to determine whether CPU_CLK clock source from either CG_PLL or external clock chip. 1=Invert the strap that controls CPU_CLK clock source.																				
9:8	Reserved.																				
7:4	<b>SLT_GFX_CLK_ClockRequestMapping.</b> Read-write. Reset: Fh. Gfx PCIe clock pins (SLT_GFX_CLKP/SLT_GFX_CLKN) output control. SLT_GFX_CLKP/SLT_GFX_CLKN pins are powered off when FCH is strapped to use an external clock, and powered on when FCH is strapped to operate in integrated clock mode. When FCH is in integrated clock mode, Gfx PCIe clock can be powered off according to the CLK_REQ mapping table below, and the selected CLK_REQ# input can power off Gfx PCIe clock output pins if it is asserted. Gfx_CLKREQ_Mapping: <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>0000b</td> <td>Off</td> </tr> <tr> <td>0001b</td> <td>CLK_REQ0#</td> </tr> <tr> <td>0010b</td> <td>CLK_REQ1#</td> </tr> <tr> <td>0011b</td> <td>CLK_REQ2#</td> </tr> <tr> <td>0100b</td> <td>CLK_REQ3#</td> </tr> <tr> <td>1001b-1010b</td> <td>Reserved; Off</td> </tr> <tr> <td>1010b</td> <td>CLK_REQGfx#</td> </tr> <tr> <td>1110b-1011b</td> <td>Reserved; Off</td> </tr> <tr> <td>1111b</td> <td>On</td> </tr> </tbody> </table>	Bits	Definition	0000b	Off	0001b	CLK_REQ0#	0010b	CLK_REQ1#	0011b	CLK_REQ2#	0100b	CLK_REQ3#	1001b-1010b	Reserved; Off	1010b	CLK_REQGfx#	1110b-1011b	Reserved; Off	1111b	On
Bits	Definition																				
0000b	Off																				
0001b	CLK_REQ0#																				
0010b	CLK_REQ1#																				
0011b	CLK_REQ2#																				
0100b	CLK_REQ3#																				
1001b-1010b	Reserved; Off																				
1010b	CLK_REQGfx#																				
1110b-1011b	Reserved; Off																				
1111b	On																				
3:2	<b>CG1PLL_LOCK_FREQ_SEL.</b> Read-write. Reset: 0. CG1_PLL VCO centre frequency selection for calibration. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>set VCO centre at 1200 MHz</td> </tr> <tr> <td>01b</td> <td>set VCO centre at 1000 MHz</td> </tr> <tr> <td>10b</td> <td>set VCO centre at 800 MHz</td> </tr> <tr> <td>11b</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Definition	00b	set VCO centre at 1200 MHz	01b	set VCO centre at 1000 MHz	10b	set VCO centre at 800 MHz	11b	Reserved										
Bits	Definition																				
00b	set VCO centre at 1200 MHz																				
01b	set VCO centre at 1000 MHz																				
10b	set VCO centre at 800 MHz																				
11b	Reserved																				
1	<b>CG1PLL_BYPASS_CAL.</b> Read-write. Reset: 0. CG1_PLL Calibration bypass enable. 0=Always do VCO calibration after a reset. 1=Bypass VCO calibration after a reset.																				
0	<b>CG1PLL_CAL_BYPASS_REFDIV_EN.</b> Read-write. Reset: 0. CG1_PLL Calibration reference divider bypass enable. 0=Reference divider set by MISCx10[CG1PLL_CAL_REFDIV] during calibration. 1=Bypass reference divider during calibration.																				

**MISCx08 CGPLLConfig1**

Bits	Description
31:16	<b>CG1PLL_DS_Frac.</b> Read-write. Reset: 0. CG1_PLL DS fractional setting.
15:11	Reserved.
10	<b>CG2PLL_Reset.</b> Read-write; Cleared-by-hardware. Reset: 0. 1=Reset the CG2 _PLL. The bit will be cleared to 0 by hardware after reset sequence is done.
9	Reserved.



8:6	Reserved.
5	<b>Cg1PllSataClkBypass</b> . Read-write. Reset: 0. SATA clock source frequency select from CG1_PLL. SATA reference clock output from CG1_PLL has an option for either output 100Mhz or bypass CG1_PLL RefClk input. 0=PLL 100Mhz. 1=Bypass CG1_PLL RefClk input (48Mhz or 100Mhz)
4	Reserved.
3	Reserved.
2:1	Reserved.
0	<b>CG1PLL_SpreadSpectrumEnable</b> . Read-write. Reset: 0. 1=Enable CG1_PLL Spread Spectrum. 0=Disable Spread Spectrum.

**MISCx0C CGPLLConfig2**

Bits	Description												
31:25	<b>CG1PLL_PDIV_PCIECLK</b> . Read-write. Reset: 12h. CG1_PLL 100 MHz clock post divider (PCIe and SATA clock). <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Bypass</td> </tr> <tr> <td>1h</td> <td>Bypass</td> </tr> <tr> <td>7Fh-02h</td> <td>Divide by &lt;CG1PLL_PDIV_PCIECLK&gt;</td> </tr> </tbody> </table>	Bits	Definition	0h	Bypass	1h	Bypass	7Fh-02h	Divide by <CG1PLL_PDIV_PCIECLK>				
Bits	Definition												
0h	Bypass												
1h	Bypass												
7Fh-02h	Divide by <CG1PLL_PDIV_PCIECLK>												
24:18	<b>CG1PLL_PDIV_CoreCLK</b> . Read-write. Reset: 03h. CG1_PLL 400 MHz clock post divider (core clock). <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Bypass</td> </tr> <tr> <td>1h</td> <td>Bypass</td> </tr> <tr> <td>7Fh-02h</td> <td>Divide by &lt;CG1PLL_PDIV_CoreCLK&gt;</td> </tr> </tbody> </table>	Bits	Definition	0h	Bypass	1h	Bypass	7Fh-02h	Divide by <CG1PLL_PDIV_CoreCLK>				
Bits	Definition												
0h	Bypass												
1h	Bypass												
7Fh-02h	Divide by <CG1PLL_PDIV_CoreCLK>												
17:16	<b>SATA_REFIN_ClkBufDrvStr</b> . Read-write. Reset: 01b. Specifies the drive strength control for internal SATA RefClk input differential clock buffers (from CLKGEN to SATA_PHY). <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>6X driving</td> <td>10b</td> <td>10X driving</td> </tr> <tr> <td>01b</td> <td>8X driving</td> <td>11b</td> <td>12X driving</td> </tr> </tbody> </table>	Bits	Definition	Bits	Definition	00b	6X driving	10b	10X driving	01b	8X driving	11b	12X driving
Bits	Definition	Bits	Definition										
00b	6X driving	10b	10X driving										
01b	8X driving	11b	12X driving										
15:14	<b>USB2_REFIN_ClkBufDrvStr</b> . Read-write. Reset: 01b. Specifies the drive strength control for internal USB2 RefClk input differential clock buffers (from CLKGEN to USB2_PHY). <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>1.5X driving</td> <td>10b</td> <td>3X driving</td> </tr> <tr> <td>01b</td> <td>2X driving</td> <td>11b</td> <td>3.5X driving</td> </tr> </tbody> </table>	Bits	Definition	Bits	Definition	00b	1.5X driving	10b	3X driving	01b	2X driving	11b	3.5X driving
Bits	Definition	Bits	Definition										
00b	1.5X driving	10b	3X driving										
01b	2X driving	11b	3.5X driving										
13:12	<b>USB3_REFIN_ClkBufDrvStr</b> . Read-write. Reset: 01b. Specifies the drive strength control for internal USB3 RefClk input differential clock buffers (from CLKGEN to USB3_PHY). <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>2X driving</td> <td>10b</td> <td>4X driving</td> </tr> <tr> <td>01b</td> <td>3X driving</td> <td>11b</td> <td>5X driving</td> </tr> </tbody> </table>	Bits	Definition	Bits	Definition	00b	2X driving	10b	4X driving	01b	3X driving	11b	5X driving
Bits	Definition	Bits	Definition										
00b	2X driving	10b	4X driving										
01b	3X driving	11b	5X driving										
11:10	<b>PCIe_REFIN_ClkBufDrvStr</b> . Read-write. Reset: 01b. Specifies the drive strength control for internal PCIe RefClk input differential clock buffers (from CLKGEN to PCIe). <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>2X driving</td> <td>10b</td> <td>4X driving</td> </tr> <tr> <td>01b</td> <td>3X driving</td> <td>11b</td> <td>5X driving</td> </tr> </tbody> </table>	Bits	Definition	Bits	Definition	00b	2X driving	10b	4X driving	01b	3X driving	11b	5X driving
Bits	Definition	Bits	Definition										
00b	2X driving	10b	4X driving										
01b	3X driving	11b	5X driving										

9:8	<b>CGPLL_REFIN_ClkBufDrvStr.</b> Read-write. Reset: 01b. Specifies the drive strength control for CGPLL external refclk input differential clock buffers (from CLKGEN to CGPLL).												
	<table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>1.5X driving</td> <td>10b</td> <td>3X driving</td> </tr> <tr> <td>01b</td> <td>2X driving</td> <td>11b</td> <td>3.5X driving</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	<u>Bits</u>	<u>Definition</u>	00b	1.5X driving	10b	3X driving	01b	2X driving	11b	3.5X driving
<u>Bits</u>	<u>Definition</u>	<u>Bits</u>	<u>Definition</u>										
00b	1.5X driving	10b	3X driving										
01b	2X driving	11b	3.5X driving										
7:2	Reserved.												
1:0	<b>Cg1BgVrefBias.</b> Read-write. Reset: 0. Specifies the CGPLL voltage regulator current bias control from Bandgap VREG_BIAS[1:0].												
	<table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>100 uA</td> <td>10b</td> <td>75 uA</td> </tr> <tr> <td>01b</td> <td>120 uA</td> <td>11b</td> <td>85 uA</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	<u>Bits</u>	<u>Definition</u>	00b	100 uA	10b	75 uA	01b	120 uA	11b	85 uA
<u>Bits</u>	<u>Definition</u>	<u>Bits</u>	<u>Definition</u>										
00b	100 uA	10b	75 uA										
01b	120 uA	11b	85 uA										

**MISCx10 CGPLLConfig3**

Bits	Description								
31:30	Reserved.								
29:26	<b>CG1PLL_CAL_REFDIV.</b> Read-write. Reset: 0100b. Calibration reference divider control. It is only used when <code>MISCx04[CG1PLL_CAL_BYPASS_REFDIV_EN] == 0</code> . <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>invalid</td> </tr> <tr> <td>1h</td> <td>invalid</td> </tr> <tr> <td>Fh-2h</td> <td>Divide by &lt;CG1PLL_CAL_REFDIV&gt;</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	0h	invalid	1h	invalid	Fh-2h	Divide by <CG1PLL_CAL_REFDIV>
<u>Bits</u>	<u>Definition</u>								
0h	invalid								
1h	invalid								
Fh-2h	Divide by <CG1PLL_CAL_REFDIV>								
25:22	<b>CG1PLL_FBDIV_Fraction.</b> Read-write. Reset: 0. Fractional part setting of the CG1_PLL feedback divider value. <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> </tr> <tr> <td>9-1</td> <td>&lt;CG1PLL_FBDIV_Fraction*0.1&gt;</td> </tr> <tr> <td>15-10</td> <td>0</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	0	0	9-1	<CG1PLL_FBDIV_Fraction*0.1>	15-10	0
<u>Bits</u>	<u>Definition</u>								
0	0								
9-1	<CG1PLL_FBDIV_Fraction*0.1>								
15-10	0								
21:10	<b>CG1PLL_FBDIV.</b> Read-write. Reset: 50. Integer part setting of the CG1_PLL Feedback divider value. <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Divide by 2</td> </tr> <tr> <td>1</td> <td>Divide by 2</td> </tr> <tr> <td>FFFh-2h</td> <td>Divide by &lt;CG1PLL_FBDIV&gt;</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	0	Divide by 2	1	Divide by 2	FFFh-2h	Divide by <CG1PLL_FBDIV>
<u>Bits</u>	<u>Definition</u>								
0	Divide by 2								
1	Divide by 2								
FFFh-2h	Divide by <CG1PLL_FBDIV>								
9:0	<b>CG1PLL_REFDIV.</b> Read-write. Reset: 2. Reference divider control. It is only used after calibration is done. <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Bypass</td> </tr> <tr> <td>1</td> <td>Bypass</td> </tr> <tr> <td>3FFh-2h</td> <td>Divide by &lt;CG1PLL_REFDIV&gt;</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	0	Bypass	1	Bypass	3FFh-2h	Divide by <CG1PLL_REFDIV>
<u>Bits</u>	<u>Definition</u>								
0	Bypass								
1	Bypass								
3FFh-2h	Divide by <CG1PLL_REFDIV>								

**MISCx14 CGPLLConfig4**

Bits	Description

31:16	<b>CG1PLL_SS_AMOUNT_DSFRAC</b> . Read-write. Reset: 1FFFh. CG1_PLL Spread spectrum amount DS setting.
15:0	<b>CG1PLL_SS_STEP_SIZE_DSFRAC</b> . Read-write. Reset: 16Dh. CG1_PLL Spread spectrum step size DS control.

**MISCx18 CGPLLConfig5**

Bits	Description										
31:30	Reserved.										
29	<b>CG2PLL_VCO800M_EN</b> . Read-write. Reset: 0. 1=Enable CG2_PLL VCO running at 800 MHz. 0=Disable. A warm reset is required after programming this bit to enable/disable VCO running at 800 MHz.										
28:27	Reserved.										
26	<b>CG1PLL_VCO800M_EN</b> . Read-write. Reset: 0. 1=Enable CG1_PLL VCO running at 800 MHz. 0=Disable. A warm reset is required after setting this bit to enable/disable VCO running at 800 MHz.										
25:24	<b>CG1PLL_PCALREF</b> . Read-write. Reset: 0. CG1_PLL VCO input2 control. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>close loop configuration</td> </tr> <tr> <td>1</td> <td>1/3 supply voltage</td> </tr> <tr> <td>2</td> <td>1/2 supply voltage</td> </tr> <tr> <td>3</td> <td>2/3 supply voltage</td> </tr> </tbody> </table>	Bits	Definition	0	close loop configuration	1	1/3 supply voltage	2	1/2 supply voltage	3	2/3 supply voltage
Bits	Definition										
0	close loop configuration										
1	1/3 supply voltage										
2	1/2 supply voltage										
3	2/3 supply voltage										
23:22	<b>CG1PLL_PVCOREF</b> . Read-write. Reset: 0. CG1_PLL VCO input1 control. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>close loop configuration</td> </tr> <tr> <td>1</td> <td>1/3 supply voltage</td> </tr> <tr> <td>2</td> <td>1/2 supply voltage</td> </tr> <tr> <td>3</td> <td>2/3 supply voltage</td> </tr> </tbody> </table>	Bits	Definition	0	close loop configuration	1	1/3 supply voltage	2	1/2 supply voltage	3	2/3 supply voltage
Bits	Definition										
0	close loop configuration										
1	1/3 supply voltage										
2	1/2 supply voltage										
3	2/3 supply voltage										
21:13	Reserved.										
12	<b>CG1PLL_SS_MODE</b> . Read-write. Reset: 1. CG1_PLL Spread spectrum mode select. 0=Center spread. 1=Down spread.										
11:8	<b>CG1PLL_SS_AMOUNT_NFRAC_SLIP</b> . Read-write. Reset: 2. CG1_PLL Spread spectrum amount fractional setting.										
7:0	<b>CG1PLL_SS_AMOUNT_FBDIV</b> . Read-write. Reset: 0. CG1_PLL Spread spectrum amount setting.										

**MISCx1C CGPLLConfig6**

BIOS: See [2.15.9.3 \[CG\\_PLL CMOS Clock Driver Setting for Power Saving\]](#).

Bits	Description
31:30	Reserved.
29	<b>CgpllClkDriverUpdate</b> . Read-write. Cold Reset: 0. CGPLL clock output drive type update. CGPLL has CML and CMOS type of clock output drivers. The selections are in <a href="#">MISCx1C[28:21]</a> . 0=CG_PLL clock driver type selection only gets updated from <a href="#">MISCx1C[28:21]</a> when CG_PLL got reset. 1=CG_PLL clock driver type selection gets updated on the fly from <a href="#">MISCx1C[28:21]</a> .

28:25	<p><b>Cg1RefClk48MHzDriverType.</b> Read-write. Cold Reset: 0. Selects CG1_PLL RefClk 48Mhz clock driver type. For each bit, 0=CML type clock driver. 1=CMOS type clock driver (consume less power).</p> <table border="1"> <thead> <tr> <th>Bit</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>[0]</td> <td>Reserved</td> </tr> <tr> <td>[1]</td> <td>Buf48 MHz for USB2 and SATA</td> </tr> <tr> <td>[2]</td> <td>Reserved</td> </tr> <tr> <td>[3]</td> <td>Reserved</td> </tr> </tbody> </table>	Bit	Definition	[0]	Reserved	[1]	Buf48 MHz for USB2 and SATA	[2]	Reserved	[3]	Reserved
Bit	Definition										
[0]	Reserved										
[1]	Buf48 MHz for USB2 and SATA										
[2]	Reserved										
[3]	Reserved										
24:21	<p><b>Cg1ClkDriverType.</b> Read-write. Cold Reset: 0. Selects CG1_PLL clock driver type. For each bit, 0=CML type clock driver. 1=CMOS type clock driver (consume less power).</p> <table border="1"> <thead> <tr> <th>Bit</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>[0]</td> <td>USB</td> </tr> <tr> <td>[1]</td> <td>SATA</td> </tr> <tr> <td>[2]</td> <td>Reserved</td> </tr> <tr> <td>[3]</td> <td>PCIe</td> </tr> </tbody> </table>	Bit	Definition	[0]	USB	[1]	SATA	[2]	Reserved	[3]	PCIe
Bit	Definition										
[0]	USB										
[1]	SATA										
[2]	Reserved										
[3]	PCIe										
20	<p><b>SataRefClkSrc.</b> Read-write. Reset: 0. Selects SATA reference clock source. 0=SATA reference clock is from CG XTAL 48 Mhz non-spread. 1=Reserved.</p>										
19:18	Reserved.										
17:9	<p><b>CG1PLL_LF_MODE.</b> Read-write. Reset: 0FEh. CG1_PLL Loop filter control. It defines the setting of loop filter R and C values.</p>										
8	<p><b>CG2PLL_RefClk_Sel.</b> Read-write. Reset: 1. CG2_PLL reference clock select. 0=100 MHz from external clock chip. 1=25 MHz from XTAL.</p>										
7	<p><b>Force25mXtalPadPwdn.</b> Read-write. Reset: 0. Force 25M XTAL Pad Power Down. Set this bit and <a href="#">MISCx40[Pwdn25Mxtal]</a> to 1 to power down 25 MHz XTAL Pad</p>										
6	<p><b>CG2PLL_PWDN_FORCE.</b> Read-write. Reset: 0. Force CG2_PLL Power Down.</p>										
5	Reserved.										
4:0	<p><b>CG1PLL_CAL_MODE.</b> Read-write. Reset: 5. CG1_PLL VCO calibration mode control.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>0xxxxb</td> <td>VCO mode set to calibrated values</td> </tr> <tr> <td>1xxxxb</td> <td>VCO mode forced to the value set in CG1PLL_CAL_MODE[3:0]</td> </tr> </tbody> </table>	Bits	Definition	0xxxxb	VCO mode set to calibrated values	1xxxxb	VCO mode forced to the value set in CG1PLL_CAL_MODE[3:0]				
Bits	Definition										
0xxxxb	VCO mode set to calibrated values										
1xxxxb	VCO mode forced to the value set in CG1PLL_CAL_MODE[3:0]										

**MISCx20 IMPCalibration**

Bits	Description
31:26	<b>CG1PLL_SPARE.</b> Read-write. Reset: 0. CG1_PLL general spare pins.
25:23	Reserved.
22:20	<b>CG1PLL_DIG_SPARE.</b> Read-write. Reset: 0. CG1_PLL digital block spare pins.
19	<b>CG1PLL_SATA_CLK_CUSTOM_PDIV_EN.</b> Read-write. Reset: 1. CG1_PLL CLKOUT3B (SATA 100 MHz) Post-dividers selection. 0=Use Programmable CMOS PostDIV inside digital block. 1=Use Custom CMOS Post-dividers.
18	<b>CG1PLL_PCIE_CLK_CUSTOM_PDIV_EN.</b> Read-write. Reset: 1. CG1_PLL CLKOUT3A (PCIe 100 MHz) post divider selection. 0=Use Programmable CMOS PostDIV inside digital block. 1=Use Custom CMOS Post-dividers.

17	<b>CG1PLL_USB2CLK_CUSTOM_PDIV_EN.</b> Read-write. Reset: 0. CG1_PLL CLKOUT2 (USB 48 MHz) post divider selection. 0=Use Programmable CMOS PostDIV inside digital block. 1=Use Custom CMOS Post-dividers.			
16	<b>CG1PLL_CLKOUT3A_PDIVSEL.</b> Read-write. Reset: 1. CG1_PLL PCIe clock select between Ref-Clk (input of reference divider) and post divider 3. 0=Input of reference divider. 1=PDIV3 Output.			
15	<b>CG1PLL_CLKOUT2_PDIVSEL.</b> Read-write. Reset: 1. CG1_PLL USB2 clock select between Ref-Clk (input of reference divider) and post divider 2. 0=Input of reference divider. 1=PDIV2 Output.			
14	<b>CG1PLL_CLKOUT1_PDIVSEL.</b> Read-write. Reset: 1. CG1_PLL Core Clock select between Ref-Clk (input of reference divider) and post divider 1. 0=Input of reference divider. 1=PDIV1 Output.			
13:12	<b>CG1PLL_PFD_PULSE_SEL.</b> Read-write. Reset: 2. CG1_PLL Change reset pulse inside PFD.			
	<u>Bits</u>	<u>Definition</u>		
	00b	~110 ps		
	01b	~220 ps		
	10b	~350 ps		
	11b	~500 ps		
11:8	<b>CG1PLL_CP.</b> Read-write. Reset: 0001b. CG1_PLL charge pump control.			
	<u>Bits</u>	<u>Definition</u>	<u>Bits</u>	<u>Definition</u>
	0000b	15 uA	1000b	55 uA
	0001b	10 uA	1001b	50 uA
	0010b	5 uA	1010b	45 uA
	0011b	0 uA	1011b	40 uA
	0100b	35 uA	1100b	75 uA
	0101b	30 uA	1101b	70 uA
	0110b	25 uA	1110b	65 uA
	0111b	20 uA	1111b	60 uA
7:0	<b>CG1PLL_IBIAS.</b> Read-write. Reset: 59h. CG1_PLL current bias control.			
	<u>Bit</u>	<u>Definition</u>		
	[0]	BIAS0		
	[1]	BIAS1		
	[2]	CPTUNE0		
	[3]	CPTUNE1		
	[4]	AMPTUNE0 (SPARE)		
	[5]	AMPTUNE1 (SPARE)		
	[6]	HALFGM		
	[7]	SPARE		

**MISCx24 ClkDrvStr1**

Bits	Description
31:20	<b>CG2PLL_FBDIV.</b> Read-write. Reset: 48. Integer part setting of the CG2_PLL feedback divider value.
	<u>Bits</u>
	0
	1
	FFFh-2h
	<u>Definition</u>
	Divide by 2
	Divide by 2
	Divide by <CG2PLL_FBDIV>

19:18	<b>SLT_GFX_ClockBufferDrivingStrengthControl</b> . Read-write. Reset: 10b. Drive strength control for SLT_GFX differential clock buffers. <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>12 mA</td> </tr> <tr> <td>01b</td> <td>13 mA</td> </tr> <tr> <td>10b</td> <td>14 mA</td> </tr> <tr> <td>11b</td> <td>15 mA</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	00b	12 mA	01b	13 mA	10b	14 mA	11b	15 mA
<u>Bits</u>	<u>Definition</u>										
00b	12 mA										
01b	13 mA										
10b	14 mA										
11b	15 mA										
17	<b>CG2PLL_BYPASS_CAL</b> . Read-write. Reset: 0. CG2_PLL Calibration bypass enable. 0=Always do VCO calibration after a reset. 1=Bypass VCO calibration after a reset.										
16	<b>CG2PLL_CAL_BYPASS_REFDIV_EN</b> . Read-write. Reset: 0. CG2_PLL Calibration reference divider bypass enable. 0=Reference divider set by MISCx28[CG2PLL_CAL_REFDIV] during calibration. 1=Bypass reference divider during calibration.										
15:12	<b>CG2PLL_VREG_BIAS</b> . Read-write. Reset: 0. CG2_PLL Voltage regulator current bias control (Spare).										
11:10	<b>CG2PLL_VREG_CNTL</b> . Read-write. Reset: 0. CG2_PLL Voltage regulator voltage control (Spare).										
9	<b>CG2PLL_VREG_VREF_SEL</b> . Read-write. Reset: 0. CG2_PLL Select Voltage regulator reference (Spare). 0=Bandgap. 1=Resistor Divider.										
8	<b>CG2PLL_PLLCORE_PWDN_EN</b> . Read-write. Reset: 0. CG2_PLL Control for power-down PLL's analog core (including PFD/CP, OTA, VCO, REFGENS, LF and all Diff output drivers). It is only used for external clock-chip mode to bypass XTALIN 25 MHz to SATA differentially when CG2_PLL is required to be off. 0=Disable. 1=Enable.										
7:6	<b>GppClk3_ClockBufferDrivingStrengthControl</b> . Read-write. Reset: 10b. Drive strength control for GPP_CLK3 differential clock buffers. <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>12 mA</td> </tr> <tr> <td>01b</td> <td>13 mA</td> </tr> <tr> <td>10b</td> <td>14 mA</td> </tr> <tr> <td>11b</td> <td>15 mA</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	00b	12 mA	01b	13 mA	10b	14 mA	11b	15 mA
<u>Bits</u>	<u>Definition</u>										
00b	12 mA										
01b	13 mA										
10b	14 mA										
11b	15 mA										
5:4	<b>GppClk2_ClockBufferDrivingStrengthControl</b> . Read-write. Reset: 10b. Drive strength control for GPP_CLK2 differential clock buffers. <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>12 mA</td> </tr> <tr> <td>01b</td> <td>13 mA</td> </tr> <tr> <td>10b</td> <td>14 mA</td> </tr> <tr> <td>11b</td> <td>15 mA</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	00b	12 mA	01b	13 mA	10b	14 mA	11b	15 mA
<u>Bits</u>	<u>Definition</u>										
00b	12 mA										
01b	13 mA										
10b	14 mA										
11b	15 mA										

3:2	<b>GppClk1_ClockBufferDrivingStrengthControl.</b> Read-write. Reset: 10b. Drive strength control for GPP_CLK1 differential clock buffers. <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>12 mA</td> </tr> <tr> <td>01b</td> <td>13 mA</td> </tr> <tr> <td>10b</td> <td>14 mA</td> </tr> <tr> <td>11b</td> <td>15 mA</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	00b	12 mA	01b	13 mA	10b	14 mA	11b	15 mA
<u>Bits</u>	<u>Definition</u>										
00b	12 mA										
01b	13 mA										
10b	14 mA										
11b	15 mA										
1:0	<b>GppClk0_ClockBufferDrivingStrengthControl.</b> Read-write. Reset: 10b. Drive strength control for GPP_CLK0 differential clock buffers. <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>12 mA</td> </tr> <tr> <td>01b</td> <td>13 mA</td> </tr> <tr> <td>10b</td> <td>14 mA</td> </tr> <tr> <td>11b</td> <td>15 mA</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	00b	12 mA	01b	13 mA	10b	14 mA	11b	15 mA
<u>Bits</u>	<u>Definition</u>										
00b	12 mA										
01b	13 mA										
10b	14 mA										
11b	15 mA										

**MISCx28 ClkDrvStr2**

Bits	Description										
31:28	<b>CG2PLL_FBDIV_FRACTION.</b> Read-write. Reset: 0. Fractional part setting of CG2_PLL Feed-back divider value. <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> </tr> <tr> <td>9-1</td> <td>&lt;CG2PLL_FBDIV_FRACTION*0.1&gt;</td> </tr> <tr> <td>15-10</td> <td>0</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	0	0	9-1	<CG2PLL_FBDIV_FRACTION*0.1>	15-10	0		
<u>Bits</u>	<u>Definition</u>										
0	0										
9-1	<CG2PLL_FBDIV_FRACTION*0.1>										
15-10	0										
27:24	Reserved.										
23:22	<b>CG2PLL_FBDIV_FRACTION_CTL.</b> Read-write. Reset: 1. CG2_PLL Slip Req/Ack handshaking control. <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Slip_req/Slip_ack handshake (8 VCO clock cycles per slip)</td> </tr> <tr> <td>01b</td> <td>Slip once every clock cycle (ignore slip_ack)</td> </tr> <tr> <td>10b</td> <td>Slip once every 2 clock cycles (ignore slip_ack)</td> </tr> <tr> <td>11b</td> <td>Slip once every 4 clock cycles (ignore slip_ack)</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	00b	Slip_req/Slip_ack handshake (8 VCO clock cycles per slip)	01b	Slip once every clock cycle (ignore slip_ack)	10b	Slip once every 2 clock cycles (ignore slip_ack)	11b	Slip once every 4 clock cycles (ignore slip_ack)
<u>Bits</u>	<u>Definition</u>										
00b	Slip_req/Slip_ack handshake (8 VCO clock cycles per slip)										
01b	Slip once every clock cycle (ignore slip_ack)										
10b	Slip once every 2 clock cycles (ignore slip_ack)										
11b	Slip once every 4 clock cycles (ignore slip_ack)										
21:19	<b>OSCOUT2_CLK_SEL.</b> Read-write. Reset: 0. Defines auxiliary output clock frequency on USB-CLK/OSCOUT2 pin. <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>48 MHz, non-spread</td> </tr> <tr> <td>001b</td> <td>25 MHz, spread</td> </tr> <tr> <td>111b-010b</td> <td>Reserved</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	000b	48 MHz, non-spread	001b	25 MHz, spread	111b-010b	Reserved		
<u>Bits</u>	<u>Definition</u>										
000b	48 MHz, non-spread										
001b	25 MHz, spread										
111b-010b	Reserved										
18:16	<b>OSCOUT1_CLK_SEL.</b> Read-write. Reset: 0. Defines auxiliary output clock frequency on OSCOUT1 pin. <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>48 MHz, non-spread</td> </tr> <tr> <td>001b</td> <td>25 MHz, spread</td> </tr> <tr> <td>111b-010b</td> <td>Reserved</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	000b	48 MHz, non-spread	001b	25 MHz, spread	111b-010b	Reserved		
<u>Bits</u>	<u>Definition</u>										
000b	48 MHz, non-spread										
001b	25 MHz, spread										
111b-010b	Reserved										

15:14	<b>CG2PLL_LOCK_FREQ_SEL.</b> Read-write. Reset: 0. CG2_PLL VCO centre frequency selection for calibration. <u>Bits</u> 00b 01b 10b 11b	<u>Definition</u> set VCO centre at 1200 MHz set VCO centre at 1000 MHz set VCO centre at 800 MHz Forbidden
13:4	<b>CG2PLL_REF_DIV.</b> Read-write. Reset: 1. CG2_PLL Reference divider control. It is only used after calibration is done. <u>Bits</u> 0 1 3Fh-2h	<u>Definition</u> Bypass Bypass Divide by <CG2PLL_REF_DIV>
3:0	<b>CG2PLL_CAL_REFDIV.</b> Read-write. Reset: 2. CG2_PLL Calibration reference divider control. It is only used when <code>MISCx24[CG2PLL_CAL_BYPASS_REFDIV] == 0</code> . <u>Bits</u> 0 1 Fh-2h	<u>Definition</u> Invalid Invalid Divide by <CG2PLL_CAL_REFDIV>

**MISCx2C ClkGatedCntl**

Bits	Description	
31:28	Reserved.	
27:26	<b>CG2PLL_PFD_PULSE_SEL.</b> Read-write. Reset: 2. CG2_PLL Change reset pulse inside PFD. <u>Bits</u> 00b 01b 10b 11b	<u>Definition</u> ~110 ps ~220 ps ~350 ps ~500 ps
25:18	<b>CG2PLL_IBIAS.</b> Read-write. Reset: 59h. CG2_PLL Current bias control. <u>Bit</u> [0] [1] [2] [3] [4] [5] [6] [7]	<u>Definition</u> BIAS0 BIAS1 CPTUNE0 CPTUNE1 AMPTUNE0 --SPARE AMPTUNE1 --SPARE HALFGM SPARE
17	<b>BlinkClkGateOffEn: B-Link Clock Gate Off Enable.</b> Read-write. Cold Reset: 0. BIOS: See <a href="#">2.15.9.2</a> . Internal B-Link clock has two clock trees: one is a free running clock and the other is a gated clock. When all controllers agree to stop the gated B-Link clock and this bit is set, clock gating logic will gate off the clock tree from clock root. 0=Disable BLink Clock Gate Off function. 1=Enable BLink Clock Gate Off function. This is a sticky bit.	



16	<b>AlinkClkGateOffEn: A-Link Clock Gate Off Enable.</b> Read-write. Cold Reset: 0. BIOS: See 2.15.9.2. Internal A-Link clock has two clock trees: one is a free-running clock and the other is a gated clock. When all controllers agree to stop the gated A-Link clock and this bit is set, clock gating logic will gate off the clock tree from clock root. 0=Disable A-Link Clock Gate-Off function. 1=Enable A-Link Clock Gate-Off function. This is a sticky bit.						
15:8	<b>BlinkClkGateOffThreshold.</b> Read-write. Reset: 0. B-Link Clock Gate Off Threshold. When all controllers agree to stop B-Link clock, clock gating logic will start a timer and deassert BLCIk_Enable when the timer reaches a programmable threshold (BLCIk_Enable is an internal handshake signal indicating whether gated B-Link clock is running or not). <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>00h</td> <td>Gated clock will stop at 3 falling edge after BLCIk_Enable is deasserted.</td> </tr> <tr> <td>FFh-01h</td> <td>Gated clock will stop at &lt;3+AlinkClkGateOffThreshold&gt; falling edge after BLCIk_Enable is deasserted.</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	00h	Gated clock will stop at 3 falling edge after BLCIk_Enable is deasserted.	FFh-01h	Gated clock will stop at <3+AlinkClkGateOffThreshold> falling edge after BLCIk_Enable is deasserted.
<u>Bits</u>	<u>Definition</u>						
00h	Gated clock will stop at 3 falling edge after BLCIk_Enable is deasserted.						
FFh-01h	Gated clock will stop at <3+AlinkClkGateOffThreshold> falling edge after BLCIk_Enable is deasserted.						
7:0	<b>AlinkClkGateOffThreshold.</b> Read-write. Reset: 0. A-Link Clock Gate Off Threshold. When all controllers agree to stop A-Link clock, clock gating logic will start a timer and deassert ALClk_Enable when the timer reaches a programmable threshold (ALClk_Enable is an internal handshake signal indicating whether gated A-Link clock is running or not). <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>00h</td> <td>Gated clock will stop at 3 falling edge after ALClk_Enable is deasserted.</td> </tr> <tr> <td>FFh-01h</td> <td>Gated clock will stop at &lt;3+AlinkClkGateOffThreshold&gt; falling edge after ALClk_Enable is deasserted.</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	00h	Gated clock will stop at 3 falling edge after ALClk_Enable is deasserted.	FFh-01h	Gated clock will stop at <3+AlinkClkGateOffThreshold> falling edge after ALClk_Enable is deasserted.
<u>Bits</u>	<u>Definition</u>						
00h	Gated clock will stop at 3 falling edge after ALClk_Enable is deasserted.						
FFh-01h	Gated clock will stop at <3+AlinkClkGateOffThreshold> falling edge after ALClk_Enable is deasserted.						

**MISCC30 CGPLLConfig7**

Bits	Description										
31:30	<b>CG2PLL_DS_ORDER.</b> Read-write. Reset: 0. CG2_PLL DS order setting. <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>disable</td> </tr> <tr> <td>1</td> <td>1st order</td> </tr> <tr> <td>2</td> <td>2nd order</td> </tr> <tr> <td>3</td> <td>3rd order</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	0	disable	1	1st order	2	2nd order	3	3rd order
<u>Bits</u>	<u>Definition</u>										
0	disable										
1	1st order										
2	2nd order										
3	3rd order										
29	<b>CG2PLL_SS_MODE.</b> Read-write. Reset: 1. CG2_PLL Spread spectrum mode select. 0=Centre spread. 1=Down spread.										
28	<b>CG2PLL_DS_PRBS_EN.</b> Read-write. Reset: 0. Enable CG2_PLL PRBS17 generator in DS modulator.										
27:21	Reserved.										
20:14	<b>CG2_PDIV_PCIECLK.</b> Read-write. Reset: 12. CG2_PLL 100 MHz clock post divider (PCIe and SATA clock). <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Bypass</td> </tr> <tr> <td>1</td> <td>Bypass</td> </tr> <tr> <td>7Fh-2h</td> <td>Divide by &lt;CG2_PDIV_PCIECLK&gt;</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	0	Bypass	1	Bypass	7Fh-2h	Divide by <CG2_PDIV_PCIECLK>		
<u>Bits</u>	<u>Definition</u>										
0	Bypass										
1	Bypass										
7Fh-2h	Divide by <CG2_PDIV_PCIECLK>										

13:7	<b>CG2_PDIV_USB2CLK.</b> Read-write. Reset: 25. CG2_PLL USB2 clock post divider (USB 2 48 MHz clock). <u>Bits</u> 0 1 7Fh-2h <u>Definition</u> Bypass Bypass Divide by <CG2_PDIV_USB2CLK>
6:0	<b>CG2_PDIV_CoreCLK.</b> Read-write. Reset: 3. CG2_PLL 400 MHz clock post divider (core clock). <u>Bits</u> 0 1 7Fh-2h <u>Definition</u> Bypass Bypass Divide by <CG2_PDIV_CoreCLK>

**MISCx34 CGPLLConfig8**

Bits	Description
31:16	<b>CG2PLL_SS_STEP_SIZE_DSFRAC.</b> Read-write. Reset: 0. CG2_PLL Spread spectrum step size DS control.
15:0	<b>CG2PLL_DS_FRAC.</b> Read-write. Reset: 0. CG2_PLL DS fractional setting.

**MISCx38 CGPLLConfig9**

Bits	Description
31	Reserved.
30	<b>CG2PLL_CLKOUT3A_PDIVSEL.</b> Read-write. Reset: 1. CG2_PLL CLKOUT3A select between RefClk (input of reference divider) and post divider 3. 0=Input of reference divider. 1=PDIV3 Output.
29	<b>CG2PLL_CLKOUT2_PDIVSEL.</b> Read-write. Reset: 1. CG2_PLL CLKOUT2 select between Ref-Clk (input of reference divider) and post divider 2. 0=Input of reference divider. 1=PDIV2 Output.
28	<b>CG2PLL_CLKOUT1_PDIVSEL.</b> Read-write. Reset: 1. CG2_PLL CLKOUT1 select between Ref-Clk (input of reference divider) and post divider 1. 0=Input of reference divider. 1=PDIV1 Output.
27:24	<b>CG2PLL_SS_AMOUNT_NFRAC_SLIP.</b> Read-write. Reset: 0. CG2_PLL spread spectrum amount fractional setting.
23:16	<b>CG2PLL_SS_AMOUNT_FBDIV.</b> Read-write. Reset: 0. CG2_PLL spread spectrum amount setting.
15:0	<b>CG2PLL_SS_AMOUNT_DSFRAC.</b> Read-write. Reset: 0. CG2_PLL spread spectrum amount DS setting.

**MISCx3C CGPLLConfig10**

Bits	Description
31	Reserved.
30:27	<b>CG2PLL_CP.</b> Read-write. Reset: 2. CG2_PLL Charge pump control. <u>Bits</u> 0000b 1111b-0001b <u>Definition</u> 0uA <CG2PLL_CP*5uA>

26:18	<b>CG2PLL_LF_MODE.</b> Read-write. Reset: DEh. CG2_PLL Loop filter control. It defines the setting of loop filter R and C values.										
17:13	<b>CG2PLL_CAL_MODE.</b> Read-write. Reset: 5. CG2_PLL VCO calibration mode control. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>0xxxxb</td> <td>VCO mode set to calibrated values</td> </tr> <tr> <td>1xxxxb</td> <td>VCO mode forced to the value set in CORE_CG2PLL_CAL_MODE[3:0]</td> </tr> </tbody> </table>	Bits	Definition	0xxxxb	VCO mode set to calibrated values	1xxxxb	VCO mode forced to the value set in CORE_CG2PLL_CAL_MODE[3:0]				
Bits	Definition										
0xxxxb	VCO mode set to calibrated values										
1xxxxb	VCO mode forced to the value set in CORE_CG2PLL_CAL_MODE[3:0]										
12:11	<b>CG2PLL_PCALREF.</b> Read-write. Reset: 0. CG2_PLL VCO input2 control. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>close loop configuration</td> </tr> <tr> <td>1</td> <td>1/3 supply voltage</td> </tr> <tr> <td>2</td> <td>1/2 supply voltage</td> </tr> <tr> <td>3</td> <td>2/3 supply voltage</td> </tr> </tbody> </table>	Bits	Definition	0	close loop configuration	1	1/3 supply voltage	2	1/2 supply voltage	3	2/3 supply voltage
Bits	Definition										
0	close loop configuration										
1	1/3 supply voltage										
2	1/2 supply voltage										
3	2/3 supply voltage										
10:9	<b>CG2PLL_PVCOREF.</b> Read-write. Reset: 0. CG2_PLL VCO input1 control. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>close loop configuration</td> </tr> <tr> <td>1</td> <td>1/3 supply voltage</td> </tr> <tr> <td>2</td> <td>1/2 supply voltage</td> </tr> <tr> <td>3</td> <td>2/3 supply voltage</td> </tr> </tbody> </table>	Bits	Definition	0	close loop configuration	1	1/3 supply voltage	2	1/2 supply voltage	3	2/3 supply voltage
Bits	Definition										
0	close loop configuration										
1	1/3 supply voltage										
2	1/2 supply voltage										
3	2/3 supply voltage										
8:0	Reserved.										

**MISCx40 MiscClkCntl1**

Bits	Description
31	<b>CG2_Atomic_Update.</b> Read-write; cleared-by-hardware. Reset: 0. 1=Request CG2_PLL to load FBDIV, FBDIV_Fraction, DS_FRAS value into CG2_PLL. The bit is cleared to 0 by hardware after request send to CG2_PLL.
30	<b>CG1_Atomic_Update.</b> Read-write; cleared-by-hardware. Reset: 0. 1=Request CG1_PLL to load FBDIV, FBDIV_Fraction, DS_FRAS value into CG1_PLL. The bit is cleared to 0 by hardware after request send to CG1_PLL.
29	<b>Atomic_Update_R_Skip.</b> Read-write. Reset: 0. 1=Hardware initializes CG1 and CG2 atomic update regardless whether CG_PLL is ready from previous update.
28:27	<b>SDCLK_PDIVSEL.</b> Read-write. Reset: 0. SDCLK Post divider selection.
26	<b>CG1PLL_FBDIV_Test.</b> Read-write. Reset: 0. 1=Enable loading MISCx10[CG1PLL_FBDIV] value for testing. A warm reset is required for this bit to take effect.
25:21	Reserved.
20	<b>EnableBlinkClkSlowMode.</b> Read-write. Reset: 0. 1=Enable internal core clock (B-Link clock) running at slower speed (66 MHz) for power saving. This bit only takes effect when MISCx40[1]== 1.
19	<b>DrvStrAuxiliaryClk2.</b> Read-write. Reset: 1. Drive Strength Control for Auxiliary Clock2 (USB-CLK/14M_25M_48M_50M_OSC). 0=4 mA. 1=8 mA.
18	<b>DrvStrAuxiliaryClk1.</b> Read-write. Reset: 1. Drive Strength Control for Auxiliary Clock1 (14M_25M_48M_50M_OSC). 0=4 mA. 1=8 mA.

17	<b>Pwdn25Mxtal.</b> Read-write. Reset: 0. 1=Power off 25M Xtal. 0=Power on 25M Xtal. Setting this bit to 1 can power off 25 MHz Xtal pad when the following conditions are not true. <ul style="list-style-type: none"> <li>• chip in integrated clock mode.</li> <li>• USB 48 MHz clock from CG2_PLL.</li> <li>• SATA use CG2_PLL clock as ref clock source.</li> <li>• 14.318 MHz clock generated from CG2_PLL.</li> <li>• 25 MHz Auxiliary clock selected.</li> </ul>
16	Reserved.
15	<b>BlinkClkSlowEnable.</b> Read-write. Reset: 0. 1=Internal B-Link clock runs on 66 MHz. The normal internal B-Link clock runs on 133 MHz. See more detail on <a href="#">MISCx40[20]</a> .
14	<b>OscClkSwitchEn.</b> Read-write. Reset: 0. BIOS: 1. 1=The FCH uses the internal PLL to generate the 14 MHz OscClk.
13	<b>InvertTermResistor.</b> Read-write. Reset: 0. 1=Invert normal RefClk (NB_PCIE_CLK) termination. 0=Normal RefClk termination.
12	<b>Auxiliary_14mClk_Sel.</b> Read-write. Reset: 0. 14 MHz clock selection for Auxiliary clock output. 0=14.285 MHz clock is from CG2_PLL. 1=14.318 MHz clock is from CLK_REQG#/14M_OSC/GPIO65 pin.
11:8	Reserved.
7	<b>OSCOUT2_OutOff.</b> Read-write. Reset: 1. If USB uses internal 48 MHz as clock source, setting this bit to 1 will turn off USBCLK/OSCOUT2 clock output. 0=OSCOUT2 pin output enable. 1=OSCOUT2 pin output disable.
6	<b>UsbClkCfg.</b> Read-write. Reset: 1. Defines whether USB uses the internal or external 48 MHz clock. 0=External 48 MHz. 1=Internal 48 MHz. If USB uses internal 48 MHz as clock source, USBCLK/OSCOUT2 pin can output 24 or 48 MHz clock depending on OSCOUT2_sel setting. If USB uses external 48 MHz clock as clock source, USBCLK/OSCOUT2 pin cannot be used as auxiliary clock output. Output enable of OSCOUT2 pin is controlled by [UsbClkCfg] and [OSCOUT2_OutOff]. When [UsbClkCfg]=1 & [OSCOUT2_OutOff]=0, pin output enable will be on.
5	Reserved.
4	<b>Usb3RefclkSel.</b> Read-write. Reset: 1. BIOS: 0. USB3.0 reference clock source selection in internal clock mode. 0=100 MHz spread clock from internal CG1_PLL. 1=100 MHz non-spread clock from internal CG2_PLL.
3	Reserved.
2	<b>OSCOUT1_ClkOutputEnb.</b> Read-write. Reset: 1. 0=Enable Auxiliary Clock1 and OSCOUT1 clock output. 1=Disable.
1	<b>CoreSpeedMode.</b> Read-write. Reset: 0. Slow down Internal Core Clock (B-Link clock) for power saving. 0=Full speed B-Link clock. 1=Slow speed B-Link clock.
0	Reserved.

**MISCx44 MiscClkCntl2**

Bits	Description
31:20	Reserved.

19	<b>CG2PLL_CLKOUT3B_CUSTOM_PDIV_EN.</b> Read-write. Reset: 0. CG2_PLL CLKOUT3B (SATA 100 MHz) Post-dividers selection. 0=Use Programmable CMOS PostDIV inside digital block. 1=Use Custom CMOS Post-dividers.
18	<b>CG2PLL_CLKOUT3A_CUSTOM_PDIV_EN.</b> Read-write. Reset: 0. CG2_PLL CLKOUT3A (PCIe 100 MHz) Post-dividers selection. 0=Use Programmable CMOS PostDIV inside digital block. 1=Use Custom CMOS Post-dividers.
17	<b>CG2PLL_CLKOUT2_CUSTOM_PDIV_EN.</b> Read-write. Reset: 0. CG2_PLL CLKOUT2 (USB2 48 MHz) Post-dividers selection. 0=Use Programmable CMOS PostDIV inside digital block. 1=Use Custom CMOS Post-dividers.
16:10	Reserved.
9	<b>CG1PLL_USB2_CLKOUT_EN.</b> Read-write. Reset: 0. CG1_PLL USB2 28 MHz Clock Output Enable.
8	<b>CG2_PLL_REFCLK_RESET.</b> Read-write. Reset: 0.
7	<b>CG1_PLL_REFCLK_RESET.</b> Read-write. Reset: 0.
6	<b>CG2PLL_ANTIGLITCH_RESET.</b> Read-write. Reset: 0. Anti-glitch reset: Forced to 1 during CG2 PLL reset and set to 0 after reset.
5	<b>CG1PLL_ANTIGLITCH_RESET.</b> Read-write. Reset: 0. Anti-glitch reset: Forced to 1 during CG1 PLL reset and set to 0 after reset.
4	<b>CG2_CAL_REFDIV_Update.</b> Read-write. Reset: 0. 1=Update CG2_CAL_REFDIV from Register <a href="#">MISCx28[CG2PLL_CAL_REFDIV]</a> . A warm reset is required for this bit to take effect.
3	<b>CG1_CAL_REFDIV_Update.</b> Read-write. Reset: 0. 1=Update CG1_CAL_REFDIV from Register <a href="#">MISCx10[CG1PLL_CAL_REFDIV]</a> . A warm reset is required for this bit to take effect.
2	<b>CG2_FBDIV_Test.</b> Read-write. Reset: 0. Enable loading <a href="#">MISCx24[CG2PLL_FBDIV]</a> value for testing. A warm reset is required for this bit to take effect. 0=Disable. 1=Enable.
1	<b>CG2_FBDIV_LoadEn.</b> Read-write. Reset: 0. 1=Enable loading CG2PLL_FBDIV value form register <a href="#">MISCx24[CG2PLL_FBDIV]</a> .
0	<b>CG1_FBDIV_LoadEn.</b> Read-write. Reset: 0. 1=Enable loading CG1PLL_FBDIV value from register <a href="#">MISCx10[CG1PLL_FBDIV]</a> .

**MISCx48 MiscClkCntl3**

Bits	Description
31:30	<b>CG2PLL_CLKOUT3B_DIFF_GAIN.</b> Read-write. Reset: 2. Diff CML Driver Strength Control for CG2PLL_CLKGEN_SATA_P/N.
29:28	<b>Cg2PllXtalDpllDiffGain.</b> Read-write. Reset: 10b. Diff CML driver strength control for XTAL_DPLL_REFCLK_P/N (48Mhz).
27:26	<b>CG2PLL_CLKOUT3A_DIFF_PRE_GAIN.</b> Read-write. Reset: 3. Diff CML Pre-Driver Strength Control for CG2PLL_DPLL_REFCLK_P/N and CG2PLL_CLKGEN_SATA_P/N.
25:24	<b>CG2PLL_CLKOUT3A_DIFF_GAIN.</b> Read-write. Reset: 2. Diff CML Driver Strength Control for CG2PLL_CLKGEN_PCIE_USB3_P/N.
23:22	<b>CG2PLL_CLKOUT2_DIFF_GAIN.</b> Read-write. Reset: 2. Diff CML Driver Strength Control for CG2PLL_CLKGEN_USB2_P/N.
21:20	<b>Cg1PllSataUsb2DiffGain.</b> Read-write. Reset: 10b. Diff CML driver strength control for CG1PLL_SATA_USB2_P/N (48Mhz).

19:16	Reserved.
15:14	<b>CG1PLL_CLKOUT3A_DIFF_GAIN</b> . Read-write. Reset: 2. Diff CML Driver Strength Control for CG1PLL_CLKGEN_PCIE_CPU_USB3_P/N
13:12	<b>CG1PLL_CLKOUT2_DIFF_GAIN</b> . Read-write. Reset: 2. Diff CML Driver Strength Control for CG1PLL_CLKGEN_USB2_P/N.
11:6	<b>CG2PLL_SPARE</b> . Read-write. Reset: 0. CG2_PLL general spare pins.
5	Reserved.
4:3	<b>Cg1XtalCmlPreDiffGain</b> . Read-write. Reset: 10b. CG1_PLL CML driver strength selection for both SATA2/USB and DPLL_REFCLK output drivers.
2:0	<b>CG2PLL_DIG_SPARE</b> . Read-write. Reset: 0. CG2_PLL spare pins to digital block.

#### MISCx4C MiscClkCntl4

Bits	Description
31	<b>LowPowerDisplay400MClkEnB</b> . Read-write. Reset: 0. Non Sticky bit. 0=Enable low power display 400Mhz clock output. 1=Disable.
30	<b>LowPowerDisplay300MClkEnB</b> . Read-write. Reset: 0. Non Sticky bit. 0=Enable low power display 300Mhz clock output. 1=Disable.
29:0	Reserved.

#### MISCx60 IdleCntrl

Bits	Description										
31:24	<b>IdleCount</b> . Read-write. Reset: X. This returns the idle count from the latest monitored period.										
23:16	Reserved.										
15:8	<b>IdleThreshold</b> . Read-write. Reset: 80h. This defines the idle-ness threshold when the dynamic clock logic down shifts the clock frequency. Each unit represents 1/256 of the interval.										
7:6	Reserved.										
5	<b>DFSslowSpeedSel</b> . Read-write. Reset: 0. Dynamic frequency Slow Speed selection. It selects clock speed to run at either 50% of full speed or 25% of full speed. 0=50% of full speed. 1=25% of full speed.										
4	<b>ForceSlowFreq</b> . Read-write. Reset: 0. 1=Force core logic clock (Secondary PCI, internal A-link, and B-link clock frequency) to run at 50% or 25% of full speed as defined by [DFSslowSpeedSel] when [IdleMonEn] is 0.										
3:2	<b>IdlePeriodSel</b> . Read-write. Reset: 1. Select interval for monitoring.										
	<table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>53.76 us</td> </tr> <tr> <td>01b</td> <td>302 us</td> </tr> <tr> <td>10b</td> <td>2.29 ms</td> </tr> <tr> <td>11b</td> <td>18.3 ms</td> </tr> </tbody> </table>	Bits	Definition	00b	53.76 us	01b	302 us	10b	2.29 ms	11b	18.3 ms
Bits	Definition										
00b	53.76 us										
01b	302 us										
10b	2.29 ms										
11b	18.3 ms										

1	<b>IdleMonCEn.</b> Read-write. Reset: 0. 1=The dynamic clock frequency logic only considers the chip to be in idle state when CPU is in the C state.
0	<b>IdleMonEn.</b> Read-write. Reset: 0. Control the dynamic clock frequency logic. When the dynamic clock frequency logic is enabled, it will monitor the amount of idle-ness defined by [IdleThreshold] within a period defined by [IdlePeriodSel]. Whenever the condition is met, it will down shift the internal clock (Secondary PCI, A-link, and B-link clock frequency) as defined by [DFSLowSpeedSel]. 0=Disable. 1=Enable.

### MISCx68 Memory Power Saving Control

Bits	Description
31:30	Reserved.
29:28	Reserved.
27:20	Reserved.
19	Reserved.
18:17	Reserved.
16	Reserved.
15:5	Reserved.
4	Reserved.
3	Reserved.
2	<b>ABBypassMemDsd.</b> Read-write. Reset: 1. BIOS: See 2.15.11.2. AB memory BypassMemdsd control. 0=Enable memory deep sleep and shutdown features. 1=Disable memory deep sleep and shutdown.
1:0	Reserved.

### MISCx70 OscFreqCounter

Bits	Description
31	<b>CountEnable.</b> Read-write. Reset: 0b. 1=Enable the internal counter to count the number of OSC clocks. When SW is not using this function, it should always set it back to 0 to conserve power.
30	<b>CountIsValid.</b> Read-only. Reset: 0b. 1=OscCountPerSec is valid. SW should always wait for this bit to be set before it reads OscCountPerSec.
29:28	Reserved.
27:0	<b>OscCountPerSec.</b> Read-write. Reset: 0000000h. Number of OSC clocks per 1 second. Whenever [CountEnable] is set, an internal counter will start counting the number of OSC clocks per second and record the count value here.

### MISCx74 HpetClkPeriod

Bits	Description
31:0	<b>HpetClkPeriod.</b> Read-write. Reset: 0429B17Eh. The register controls the value of HPETx004[CounterClkPeriod] register in HPET MMIO register space.

**MISCx78 PostCode**

Bits	Description
31:0	<b>PostCode[31:0]</b> . Read-only. Reset: 0. BIOS 32 bit writes to <b>IO080 [PostCode]</b> go to this internal 32-bit PostCode Register. Reads from <b>IO080 [PostCode]</b> return PostCode[7:0].

**MISCx80 StrapStatus**

Bits	Description
31:19	Reserved.
18	<b>BifGen2ComplianceStrap</b> . Read-only. Reset: 0. BIF gen 2 compliance strap.
17	<b>ClkGenStrap</b> . Read-only. Reset: X. BIOS: See 2.15.9. 1==Internal clocking mode; Use 48MHz crystal clock as the reference clock. 0=External clocking mode; Use 100MHz differential spread clock as the reference clock.
16	<b>BootFailTmrEnStrap</b> . Read-only. Reset: X. Enable Watchdog function.
15	<b>PciRomBootStrap</b> . Read-only. Reset: 1. PCI ROM Boot.
14:13	Reserved.
12	<b>CPUClkSelStrap</b> . Read-only. Reset: 0.
11	<b>ShortResetStrap</b> . Read-only. Reset: 0. Generate short reset
10	<b>PciPllBypStrap</b> . Read-only. Reset: 0. Bypass PCI PLL (used in functional test at tester).
9	<b>SDPllBypStrap</b> . Read-only. Reset: 0.
8	Reserved.
7	<b>I2CRomStrap</b> . Read-only. Reset: 0. Getting UMI core strap from I2C ROM or using default value.
6	Reserved.
5	Reserved.
4:3	Reserved.
2	<b>EcEnableStrap</b> . Read-only. Reset: 0. BIOS: See 2.15.8. 1=IMC is enabled.
1	<b>UseLpcRomStrap</b> . Read-only. Reset: X.
0	Reserved.

**MISCx90 AutoTransaction/Allow EC**

Bits	Description
31:10	Reserved.
9	<b>DisableAuto</b> . Read; write-1-only. Reset: 0. 1=The entire Auto Transaction logic is disabled. Once this bit is set, it cannot be cleared except by system reset.
8	<b>AllowECToAutoTransactEn</b> . Read-write. Reset: 0. Only BIOS can change this bit. 0= IMC cannot write to any of registers <b>MISCx90/MISCx94/MISCx98/MISCx9C</b> in the Auto Transaction Generation logic. 1=IMC can change any of these registers.



7:4	<b>TransactionType.</b> Read-write. Reset: 0. PCI Command type used for this auto transaction. See PCI specification for PCI command types.										
3:2	<b>ByteCount.</b> Read-write. Reset: 0. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>1 byte</td> </tr> <tr> <td>01b</td> <td>2 bytes</td> </tr> <tr> <td>10b</td> <td>4 bytes</td> </tr> <tr> <td>11b</td> <td>4 bytes</td> </tr> </tbody> </table>	Bits	Definition	00b	1 byte	01b	2 bytes	10b	4 bytes	11b	4 bytes
Bits	Definition										
00b	1 byte										
01b	2 bytes										
10b	4 bytes										
11b	4 bytes										
1	<b>DualAddr.</b> Read-write. Reset: 0. 0=Use single address cycle. 1=Use dual address cycle.										
0	<b>AutoExecute.</b> Read-write; cleared-by-hardware. Reset: 0. Writing 1 to this bit causes the HW to execute the transaction defined by bit[7:1]. Once it is written, this bit stays as 1 until the transaction is completed, in which case it will return to 0.										

**MISCx94 AutoAddrLow**

Bits	Description
31:0	<b>AutoAddrLow.</b> Read-write. Reset: 0. Low address to be used by the <a href="#">MISCx90[AutoExecute]</a> operation.

**MISCx98 AutoAddrHigh**

Bits	Description
31:0	<b>AutoAddrHigh.</b> Read-write. Reset: 0. High address to be used by the <a href="#">MISCx90[AutoExecute]</a> operation. This register is only applicable when <a href="#">MISCx90[DualAddr]</a> == 1.

**MISCx9C AutoData**

Bits	Description
31:0	<b>AutoData.</b> Read-write. Reset: 0. If <a href="#">MISCx90[TransactionType]</a> is read, this register will return the read data. If the <a href="#">MISCx90[TransactionType]</a> is a write command, this register will contain the write data. Note byte is aligned accordingly.

**MISCxC0 CPU Pstate0**

Bits	Description
31:15	Reserved.
14:12	<b>Core3Pstate.</b> Read-only. Reset: 0. See: Core0Pstate.
11	Reserved.
10:8	<b>Core2Pstate.</b> Read-only. Reset: 0. See: Core0Pstate.
7	Reserved.
6:4	<b>Core1Pstate.</b> Read-only. Reset: 0. See: Core0Pstate.

3	Reserved.
2:0	<b>Core0Pstate.</b> Read-only. Reset: 0. FCH will monitor the P state of each CPU core.
	<u>Bits</u> <u>Definition</u>
	000b                      P0
	001b                      P1
	010b                      P2
	011b                      P3
	100b                      P4
	101b                      P5
	110b                      P6
	111b                      P7

**MISCC4 CPU Pstate1**

Bits	Description
31:0	Reserved.

**MISCD0 CPU Cstate0**

Bits	Description
31:15	Reserved.
14:12	<b>Core3Cstate.</b> Read-only. Reset: 0. See: Core0Cstate.
11	Reserved.
10:8	<b>Core2Cstate.</b> Read-only. Reset: 0. See: Core0Cstate.
7	Reserved.
6:4	<b>Core1Cstate.</b> Read-only. Reset: 0. See: Core0Cstate.
3	Reserved.
2:0	<b>Core0Cstate.</b> Read-only. Reset: 0.
	<u>Bits</u> <u>Definition</u>
	000b                      C0
	001b                      C1
	010b                      C2
	011b                      C3
	100b                      C4
	101b                      C5
	110b                      C6
	111b                      C7

**MISCD4 CPU Cstate1**

Bits	Description
31:0	Reserved.

**MISCxF0 SataPortSts**

Bits	Description										
31:26	Reserved.										
25:24	<b>SataPortSel.</b> Read-write. Reset: 0. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Select “led” for Port 0 to 1</td> </tr> <tr> <td>01b</td> <td>Select “det” for Port 0 to 1</td> </tr> <tr> <td>10b</td> <td>Select “err” for Port 0 to 1</td> </tr> <tr> <td>11b</td> <td>Select “led” for Port 0 to 1</td> </tr> </tbody> </table>	Bits	Definition	00b	Select “led” for Port 0 to 1	01b	Select “det” for Port 0 to 1	10b	Select “err” for Port 0 to 1	11b	Select “led” for Port 0 to 1
Bits	Definition										
00b	Select “led” for Port 0 to 1										
01b	Select “det” for Port 0 to 1										
10b	Select “err” for Port 0 to 1										
11b	Select “led” for Port 0 to 1										
23:2	Reserved.										
1	<b>Port1Sts.</b> Read-write. Reset: X. The selected status of Port 1. This status bit indicates the internal status of SATA port 1.										
0	<b>Port0Sts.</b> Read-write. Reset: X. The selected status of Port 0. This status bit indicates the internal status of SATA port 0.										

**MISCxF4 ClkCntrlSts**

Bits	Description
31:0	Reserved.

**3.25.12 GPIO Pin control registers**

The GPIO pins are controlled by a combination of device enables and by their specific **GPIOx[E4:00]** and **IOMUXx[E4:00]** register pair.

**3.25.12.1 GPIO Registers**

GPIO register space is accessed through the AcpiMmio region, which can be memory mapped or IO mapped. The registers range from **PMx24[AcpiMmioAddr]+100h** to **PMx24[AcpiMmioAddr]+1FFh**. See **PMx24[AcpiMmioEn]**.

**GPIOx[E4:00] GPIO**

BIOS: See 2.15.6.

Bits	Description
7	<b>GpioIn.</b> Read-only; updated-by-hardware. Reset: 0. Current pin state.
6	<b>GpioOut.</b> Read-write. Reset: 0. Output state when GpioOutEnB is enabled.
5	<b>GpioOutEnB.</b> Read-write. Reset: 1. 0=Output enable. 1=Output disable.
4	<b>PullDown.</b> Read-write. Reset: 0. 0=Pull down disabled. 1=Pull down enabled.
3	<b>PullUpB.</b> Read-write. Reset: 0. 1=Pull up disabled. 0=Pull up enabled.
2	<b>Sticky.</b> Read-write. Reset: 0. 1=Bits[6:3] are sticky and can be reloaded to the default value on PCI reset. 0=Bits [6:3] can be reloaded to the default value on RsmRst and SYS_RST#. This will allow every GPIO to be sticky or non-sticky.

1	<b>OwnedByHost.</b> Read-write. Reset: 0. This bit can only be written by host (BIOS). If this bit is set, only host can write to bits[6:2] and bit 0 can no longer be written by IMC. This bit is always sticky.
0	<b>OwnedByEc.</b> Read-write. Reset: 0. This bit can only be written by IMC. If this bit is set, only IMC can write to bits[6:2] and bit 1 can no longer be written by host. This bit is always sticky.

### 3.25.12.2 IOMux Registers

IOMux register space is accessed through the AcpiMmio region, which can be memory mapped or IO mapped. The registers range from **PMx24**[AcpiMmioAddr]+D00h to **PMx24**[AcpiMmioAddr]+DFFh. See **PMx24** [AcpiMmioEn].

#### **IOMUXx[E4:00] IoMux**

Table 237: Reset Mapping for IOMUXx[E4:00]

Register	Reset	Register	Reset	Register	Reset	Register	Reset
IOMUXx[1F:00]	00h	IOMUXx3D	00h	IOMUXx60	00h	IOMUXx75	00h
IOMUXx20	00h	IOMUXx3E	00h	IOMUXx61	00h	IOMUXx76	00h
IOMUXx21	00h	IOMUXx3F	00h	IOMUXx62	02h	IOMUXx77	00h
IOMUXx[2A:22]	00h	IOMUXx40	02h	IOMUXx63	00h	IOMUXx[A0:78]	00h
IOMUXx2B	00h	IOMUXx41	00h	IOMUXx64	01h	IOMUXxA1	00h
IOMUXx2C	00h	IOMUXx42	00h	IOMUXx65	01h	IOMUXxA2	00h
IOMUXx2D	01h	IOMUXx43	00h	IOMUXx66	01h	IOMUXxA3	00h
IOMUXx2E	00h	IOMUXx44	00h	IOMUXx67	01h	IOMUXxA4	00h
IOMUXx2F	00h	IOMUXx45	00h	IOMUXx68	00h	IOMUXxA5	00h
IOMUXx30	00h	IOMUXx46	00h	IOMUXx69	00h	IOMUXxA6	00h
IOMUXx31	00h	IOMUXx47	00h	IOMUXx6A	01h	IOMUXxA7	00h
IOMUXx32	02h	IOMUXx48	00h	IOMUXx6B	01h	IOMUXxA8	00h
IOMUXx33	02h	IOMUXx49	00h	IOMUXx6C	00h	IOMUXxA9	00h
IOMUXx34	00h	IOMUXx4A	00h	IOMUXx6D	00h	IOMUXxAA	00h
IOMUXx[36:35]	00h	IOMUXx4B	00h	IOMUXx6E	00h	IOMUXx[AD:AB]	00h
IOMUXx37	02h	IOMUXx4C	00h	IOMUXx6F	00h	IOMUXxAE	00h
IOMUXx38	00h	IOMUXx4D	00h	IOMUXx70	01h	IOMUXx[B7:AF]	00h
IOMUXx39	01h	IOMUXx4E	00h	IOMUXx71	02h	IOMUXxB8	00h
IOMUXx3A	01h	IOMUXx4F	00h	IOMUXx72	00h	IOMUXx[E2:B9]	00h
IOMUXx3B	02h	IOMUXx50	00h	IOMUXx73	00h	IOMUXxE3	00h
IOMUXx3C	00h	IOMUXx[5F:51]	00h	IOMUXx74	00h	IOMUXxE4	00h

Bits	Description										
7:2	Reserved.										
1:0	<b>IOMuxGPIO.</b> Read-write. Multi-function IO pin function select for pin GPIO[X]. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr><td>00b</td><td>function 0</td></tr> <tr><td>01b</td><td>function 1</td></tr> <tr><td>10b</td><td>function 2</td></tr> <tr><td>11b</td><td>function 3</td></tr> </tbody> </table>	Bits	Definition	00b	function 0	01b	function 1	10b	function 2	11b	function 3
Bits	Definition										
00b	function 0										
01b	function 1										
10b	function 2										
11b	function 3										

### 3.25.13 Power Management (PM) Registers

PM register space is accessed through two methods:

- Indirect IO access through **IOCD6 [PM\_Index]** and **IOCD7 [PM\_Data]**. Software first programs the offset into the index register **IOCD6** and then reads from/writes to the data register **IOCD7**.
- Direct memory mapped or IO mapped access through the AcpiMmio region. The PM registers range from **PMx24[AcpiMmioAddr]+300h** to **PMx24[AcpiMmioAddr]+3FFh**. See **PMx24** for details on how to set up the AcpiMmio region.

#### PMx00 DecodeEn

Bits	Description
31	Reserved.
30	<b>Port92Enable</b> . Read-write. Reset: 1. 1=Enable the decoding of IO Port 92h.
29:28	Reserved.
27:26	Reserved.
25	<b>PMAddrEnable</b> . Read-write. Reset: 1. 1=Enable the decoding of IO Port CD6h and CD7h.
24:20	Reserved.
19	<b>RtcAddrEn</b> . Read-write. Reset: 1. 1=Enable the decoding of IO Port 70h and 71h.
18	<b>NmiAddrEn</b> . Read-write. Reset: 1. 1=Enable the decoding of IO Port 61h.
17	<b>PitAddrEn</b> . Read-write. Reset: 1. 1=Enable the decoding of IO Port 40h, 41h, 42h, 43h.
16	<b>DmaAddrEn</b> . Read-write. Reset: 1. 1=Enable the decoding of IO Port 1Fh:00h, 8Fh:80h, CFh:C0h, DFh:D0h.
15:8	Reserved.
7:6	Reserved.
5:4	Reserved.
3	<b>PmEnable</b> . Read-write. Reset: 1. 1=Enable IO port CD6h and CD7h decoding on the internal ISA bus. This bit has to be set all the time.
2	<b>TmrEnable</b> . Read-write. Reset: 1. 1=Enable 8254 timer function.
1	<b>IntrEnable</b> . Read-write. Reset: 1. 1=Enable PIC interrupt function.
0	Reserved.

#### PMx04 IsaControl

Bits	Description
31:17	Reserved.
16	<b>ABLinkClkGateEn</b> . Read-write. Reset: 0. BIOS: See 2.15.9.2 [Global A-Link / B-Link Clock Gating]. Specifies the master switch for A-link and B-link clock gating. This bit is non-sticky. 1=A-link/B-link clock gating global enable.
15:12	Reserved.
11:9	Reserved.
8	Reserved.

7	<b>DmaLimitEn.</b> Read-write. Reset: 0. 1=Enable DmaLimit on legacy DMA transfers on the LPC bus.
6:0	<b>DmaLimit.</b> Read-write. Reset: 0. Specifies the legacy DMA transfer size.

**PMx08 PciControl**

Bits	Description
31:26	Reserved.
25	<b>ForceSlpStateRetry.</b> Read-write. Reset: 0. 1=Send out SMI message before the completion response of IO writes to <code>AcpiPm1CntBlkx00[SlpTyp]</code> . This is to be used in conjunction with SMI trapping on writes to <code>AcpiPm1CntBlkx00[SlpTyp]</code> .
24	<b>ForceStpClkRetry.</b> Read-write. Reset: 1. 1=Send out STPCLK message before the completion response to the following 3 types of requests: <ul style="list-style-type: none"> <li>• IO writes to <code>AcpiPm1CntBlkx00[SlpTyp]</code>.</li> <li>• IO write LDT_STP command.</li> <li>• C1e cycle.</li> </ul> Normally it is required to send out STPCLK before completion of the cycles listed above, except for the case of SMI trapping. In the case of SMI trapping, this bit should be programmed as 0.
23	<b>AbStallEn.</b> Read-write. Reset: 0. 1=Allow the legacy DMA engine to hold the internal bus before completing legacy DMA on the LPC bus. This is only needed for certain old LPC devices.
22:20	Reserved.
19	<b>MasterNoWait.</b> Read-write. Reset: 0. 1=ACPI PCI Master doesn't wait for Slave idle when it wants to request bus. 0=PCI Master will wait for Slave idle.
18:15	Reserved.
14:12	<b>ExtIntrTime.</b> Read-write. Reset: 0. Specifies the extended interrupt time in 2 microsecond intervals. This is used for preventing APU from re-entering C state right away when it just breaks out from a C state.
11	Reserved.
10	Reserved.
9	Reserved.
8	<b>PicApicArbiter.</b> Read-write. Reset: 1. 1=Enable arbitration between PIC request and IOAPIC request.
7	<b>ForceSmafMatch.</b> Read-write. Reset: 0. 1=Enable STPGNT message matching to the expected SMAF.
6	<b>MtsAuto.</b> Read-write. Reset: 0. 1=Encode PIC interrupt request as legacy PIC ExtInt message type and PIC NMI request as legacy PIC NMI message type if IOAPIC is enabled && MtsSet==0.
5	<b>MtsSet.</b> Read-write. Reset: 1. 1=Encode PIC interrupt request as legacy PIC ExtInt message type and NMI request as legacy NMI message type. 0=Encode PIC interrupt request as ExtInt message type and NMI request as NMI message type.
4	<b>MsgIntrEnable.</b> Read-write. Reset: 0. 1=Deliver legacy PIC interrupt as message type.
3	<b>MaskMsgBmStsEn.</b> Read-write. Reset: 0. 1=Enable A20#, IGNNE#, INIT#, NMI, SMI# message delivery.
2	<b>DmaVerifyEn.</b> Read-write. Reset: 0. 1=Enable mimicking of legacy DMA VERIFY function. This is only needed for old LPC driver, such as floppy, that requires VERIFY function.

1	Reserved.
0	<b>ArbDmaDis.</b> Read-write. Reset: 0. BIOS: See 2.15.4.1. 1=Enable arbitration between legacy DMA and other PCI masters. 0=Disable arbitration between legacy DMA and other PCI masters.

### PMx0C StpClkSmaf

Bits	Description
31	Reserved.
30:28	<b>TtSmaf.</b> Read-write. Reset: 5h. Specifies the system management action field for thermal throttling STPCLK message.
27	Reserved.
26:24	<b>NsSmaf.</b> Read-write. Reset: 5h. Specifies the system management action field for normal throttling STPCLK message.
23	Reserved.
22:20	<b>S3Smaf.</b> Read-write. Reset: 4h. Specifies the system management action field for S3 STPCLK message.
19	Reserved.
18:16	<b>S1Smaf.</b> Read-write. Reset: 3h. Specifies the system management action field for S1 STPCLK message.
15	Reserved.
14:12	<b>VfSmaf.</b> Read-write. Reset: 2h. Specifies the system management action field for VFID STPCLK message.
11	Reserved.
10:8	<b>C3Smaf.</b> Read-write. Reset: 1. Specifies the system management action field for C3 STPCLK message.
7	Reserved.
6:4	<b>C2Smaf.</b> Read-write. Reset: 0. Specifies the system management action field for C2 STPCLK message.
3	Reserved.
2:0	<b>S4S5Smaf.</b> Read-write. Reset: 6h. Specifies the system management action field for S4/5 STPCLK message.

### PMx20 BiosRamEn

Bits	Description
31:8	<b>BiosRamMemAddr.</b> Read-write. Reset: FED100h. Specifies the BIOS RAM base address[31:8].
7:2	Reserved.
1	<b>BiosRamMemPmEnable.</b> Read-write. Reset: 0. 1=Enable BIOS RAM auto-deep-sleep mode.
0	<b>BiosRamMemEnable.</b> Read-write. Reset: 0. 1=Enable BIOS RAM access.

**PMx24 AcpiMmioEn**

The AcpiMmio region mapping is disabled at initial power up. Software needs to program it using the indirect IO access through the index register **IOCD6 [PM\_Index]** and the data register **IOCD7 [PM\_Data]**:

1. Select the AcpiMmio region to be memory mapped or IO mapped through **PMx24[AcpiMmioSel]**
2. Program the base address through **PMx24[AcpiMmioAddr]**.
3. Enable the direct mapping through **PMx24[AcpiMmioDecodeEn]**.

Bits	Description
31:13	<p><b>AcpiMmioAddr</b>. Read-write. Reset: 7F6C0h. Specifies the base address[31:13] of the AcpiMmio region for FCH resources. The AcpiMmio base address is FED8_0000h at default. The following lists the offsets of various register blocks within this region:</p> <p>00FFh-0000h SMBus PCI configuration registers, see <b>D14F0x00</b>.</p> <p>01FFh-0100h GPIO, see <b>GPIOx[E4:00]</b></p> <p>02FFh-0200h SMI, see <b>SMIx00</b>.</p> <p>03FFh-0300h PMIO, see <b>PMx00</b>.</p> <p>04FFh-0400h PMIO2, see <b>PM2x00</b>.</p> <p>05FFh-0500h BIOS RAM</p> <p>06FFh-0600h CMOS RAM, see <b>IO073_x00</b>.</p> <p>07FFh-0700h CMOS</p> <p>08FFh-0800h ACPI, see <b>3.25.15 [Standard ACPI Registers]</b>.</p> <p>09FFh-0900h ASF registers, see <b>ASFx00</b>.</p> <p>0AFFh-0A00h SMBus registers, see <b>SMBUSx00</b></p> <p>0BFFh-0B00h Watchdog registers, see <b>WDTx00</b>.</p> <p>0CFFh-0C00h HPET, see <b>HPETx000</b>.</p> <p>0DFFh-0D00h IOMux, see <b>IOMUXx[E4:00]</b>.</p> <p>0EFFh-0E00h Miscellaneous registers, see <b>MISCx00</b>.</p> <p>10FFh-1000h Reserved.</p> <p>1CFFh-1C00h USB3 Phy, see <b>XHCI_PMx00</b>.</p> <p>1DFFh-1D00h Wake Device (AC DC timer), see <b>AcDcTimerx00</b>.</p> <p>1EFFh-1E00h Reserved.</p>
12:2	Reserved.
1	<b>AcpiMmioSel</b> . Read-write. Reset: 0. Selects the AcpiMmio region to be in memory-mapped or IO-mapped space. 0=Memory mapped space. 1=IO mapped space.
0	<b>AcpiMmioDecodeEn</b> . Read-write. Reset: 0. BIOS: 1. 1=Enable the AcpiMmio space.

**PMx28 AsfEn**

Bits	Description
31:24	Reserved.
23	<b>AsfClkSwitchEn</b> . Read-write. Cold reset: 0. 1=Change ASF master clock from RTC (32 kHz) to the clock defined in <b>AsfClkSel</b> .



22:16	<b>AsfClkSel.</b> Read-write. Cold reset: 0. Specifies the frequency of ASF master clock.			
	<u>Bits</u>	<u>Definition</u>	<u>Bits</u>	<u>Definition</u>
	0h	~100 kHz	5h	~800 kHz
	1h	~200 kHz	6h	~900 kHz
	2h	~300 kHz	7h	~1 MHz
	3h	~400 kHz	7Fh-8h	66.67 MHz / ((AsfClkSel + 1) * 2)
	4h	~600 kHz		
15:5	<b>AsfIoBase.</b> Read-write. Cold reset: 590h. Specifies AsfIoBase[15:5]. AsfIoBase is B20h at default.			
4:3	Reserved.			
2	<b>AsfSmMasterEn.</b> Read-write. Cold reset: 0. 1=Enable ASF SMBUS master function.			
1	<b>AsfClkStretchEn.</b> Read-write. Cold reset: 0. 1=Enable clock stretching support.			
0	<b>AsfEn.</b> Read-write. Cold reset: 0. 1=Enable ASF function and IO decoding.			

**PMx2C Smbus0En**

Bits	Description
15:5	<b>SmBus0IoBase.</b> Read-write. Cold reset: 580h. Specifies Smbus0IoBase[15:5]. Smbus0IoBase is B00h at default.
4:1	Reserved.
0	<b>SmBus0En.</b> Read-write. Cold reset: 0. 1=Enable SMBUS0 function and IO decoding.

**PMx2E Smbus0Sel**

Bits	Description
7:3	Reserved.
2:1	<b>SmBus0Sel.</b> Read-write. Reset: 0. SmBus port selection. There is only one SMBus engine controlling four SMBus ports. This register routes the SMBus engine to the desired port.
	<u>Bits</u> <u>Definition</u>
	00b                          Port 0, for SMBUS on the board
	01b                          Port 2, dedicated for TSI polling
	11b-10b                      Reserved
0	Reserved.

**PMx2F SmbusSlpGate**

Bits	Description
7:6	Reserved.
5	<b>SmBusSlpGate[1].</b> Read-write. Reset:0. 1=Gate-off SmBus Port2 during S3/S5 if (PMx2E[SmBus0Sel]==01b). 0=No gate-off during S3/S5.
4	<b>SmBusSlpGate[0].</b> Read-write. Reset:0. 1=Gate-off SmBus Port0 during S3/S5 if (PMx2E[SmBus0Sel]==00b). 0=No gate-off during S3/S5.
3:0	Reserved.

**PMx34 IoApicEn**

Bits	Description
31:5	<b>IoApicBaseAddr.</b> Read-write. Reset: 7F60000h. Specifies IOAPIC base address[31:5]. IOAPIC base address[31:0] is FEC0_0000h at default.
4	<b>IoApicIdExtEn.</b> Read-write. Reset: 1. 1=Extend APIC ID defined at <a href="#">IOAPICx10_x00[ID]</a> from 4 bits to 8 bits.
3	Reserved.
2	<b>IoApicMmio.</b> Read-write. Reset: 1. 0=IO space. 1=Memory space.
1	<b>IoApicMode.</b> Read-write. Reset: 1. 1=Enable IOxAPIC. This bit is only valid if (IoApicEnable==1).
0	<b>IoApicEnable.</b> Read-write. Reset: 0. 1=Enable IOAPIC decoding.

**PMx44 BootTimerEn**

Bits	Description
31	<b>BootTmrDisable.</b> Read-write. Cold reset: 0. BIOS: 1. 0=Boot timer starts running. 1=Boot timer is stopped; it will not trigger a system reset or de-assertion on the NbPwrGood. Software should set the bit to 1 after every reset or S3/S4/S5 resume before the timer expires in 1.17 s.
30	<b>FailBootRstSts.</b> Write-1-to-clear; Read. Cold reset: 0. 0=Boot timer has not been fired. 1=Boot timer has been fired.
29	<b>ExpireBootTmr.</b> Read-write. Cold reset: 0. 1=Force boot timer to expire; then NbPwrGood can be asserted.
28	<b>BootTmrStopOnGAlink.</b> Read-write. Cold reset: 1. BIOS: 0. 0=Enable the boot timer to ensure a good boot. 1=Boot timer stops when FCH observes a good boot after PCI reset or S3/S4/S5 resume.
27	<b>BootTmrFuncEn.</b> Read-write. Cold reset: 1. BIOS: 1. 0=Disable boot timer function; avoid system restarts when performing BIOS debug. 1=Enable boot timer function; the boot timer starts to count down and will toggle NbPwrGood after 1.17 s if software has not set ExpireBootTmr to 1 after PCI reset or resuming from S3/S4/S5.
26:25	Reserved.
24:0	<b>FailBootTimer.</b> Read-write. Cold reset: 0. Specifies the counter of APU boot timer (14.318 MHz), which starts counting when both of the following conditions are met: <ul style="list-style-type: none"> <li>• BootTmrDisable==1.</li> <li>• PCI reset is not asserted.</li> </ul>

**PMx48 WatchdogTimerEn**

Bits	Description
31:3	<b>WatchdogBase.</b> Read-write. Cold reset: 1FD60000h. Specifies the watchdog timer memory mapped base address[31:3]. The base address is FEB0_0000h at default.
2	Reserved.

1	<b>WatchdogFuncDisable.</b> Read-write. Cold reset: 1. 0=Enable watchdog timer. 1=Disable watchdog timer.
0	<b>WatchdogDecodeEn.</b> Read-write. Cold reset: 0. 1=Enable decoding of watchdog timer address.

### PMx4C WatchdogTimerConfig

Bits	Description												
7:2	Reserved.												
1:0	<b>WatchdogFreq.</b> Read-write. Cold reset: 3h. Specifies the clock frequency used by the watchdog timer.												
	<table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>32 us</td> <td>10b</td> <td>100 ms</td> </tr> <tr> <td>01b</td> <td>10 ms</td> <td>11b</td> <td>1 s</td> </tr> </tbody> </table>	Bits	Definition	Bits	Definition	00b	32 us	10b	100 ms	01b	10 ms	11b	1 s
Bits	Definition	Bits	Definition										
00b	32 us	10b	100 ms										
01b	10 ms	11b	1 s										

### PMx50 HPETEn

Bits	Description
31:10	<b>HpetBaseAddress.</b> Read-write. Reset: 3FB400h. Specifies the HPET MMIO base address[31:10]. The base address is FED0_0000h at default.
9:5	Reserved.
4	<b>HpetTimer2MsiEn.</b> Read-write. Reset: 0. 1=Enable HPET timer2 MSI capability.
3	<b>HpetTimer1MsiEn.</b> Read-write. Reset: 0. 1=Enable HPET timer1 MSI capability.
2	<b>HpetTimer0MsiEn.</b> Read-write. Reset: 0. 1=Enable HPET timer0 MSI capability.
1	<b>HpetIrqEn.</b> Read-write. Reset: 0. 1=Enable HPET IRQ output.
0	<b>HpetDecodeEn.</b> Read-write. Reset: 0. 1=Enable HPET MMIO space decoding.

### PMx54 SerialIrqConfig

Bits	Description
15:14	Reserved.
13:12	Reserved.
11:8	Reserved.
7	<b>SerialIrqEnable.</b> Read-write. Cold reset: 0. 1=Enable the serial IRQ function.
6	<b>SerIrqMode.</b> Read-write. Cold reset: 0. 0=Continuous mode. 1=Active (quiet) mode.

5:2	<b>NumSerIrqBits.</b> Read-write. Cold reset: 0. Specifies the total number of serial IRQs. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>17 serial IRQs (15 IRQ#, SMI#, IOCHK#)</td> </tr> <tr> <td>1</td> <td>18 serial IRQs (15 IRQ#, SMI#, IOCHK#, INTA#)</td> </tr> <tr> <td>15-2</td> <td>&lt;NumSerIrqBits+17&gt;</td> </tr> </tbody> </table> <p>The serial IRQ can support 15 IRQ#, SMI#, IOCHK#, INTA#, INTB#, INTC#, and INTD#. When serial SMI# is used, BIOS needs to check super IO or device that generates serial SMI# for status.</p>	Bits	Definition	0	17 serial IRQs (15 IRQ#, SMI#, IOCHK#)	1	18 serial IRQs (15 IRQ#, SMI#, IOCHK#, INTA#)	15-2	<NumSerIrqBits+17>
Bits	Definition								
0	17 serial IRQs (15 IRQ#, SMI#, IOCHK#)								
1	18 serial IRQs (15 IRQ#, SMI#, IOCHK#, INTA#)								
15-2	<NumSerIrqBits+17>								
1:0	<b>NumStartBits.</b> Read-write. Cold reset: 0. This field specifies the number of clocks in the start frame. Start Frame Width = 4 + 2 * NumStartBits.								

### PMx56 RTC Control

Bits	Description
15	Reserved.
14	<b>ExtraRtcCmosEn.</b> Read-write. Cold reset: 0. 1=Software can access the extra 16 bytes of RTC CMOS RAM through <b>IO072</b> and <b>IO073</b> ; The extra RAM space are located at index 0x00-0x03, 0x40-0x43, 0x80-0x83, 0xC0-0xC3. 0=Access to the extra RTC CMOS RAM space is disabled.
13	<b>AltCmosMapEn.</b> Read-write. Cold reset: 0. 1=When accessing the RTC CMOS RAM through <b>IO070</b> and <b>IO071</b> , Bank 1 of CMOS RAM is changed: Index 0Dh:00h still returns the time and alarm settings; Index 7Fh:0Eh returns the absolute offset FFh:8Eh, which map to the extended RAM space.
12	<b>CenturyEn.</b> Read-write. Cold reset: 1. 1=Enable RTC Century support.
11	<b>MaskRtcClkOut.</b> Read-write. Cold reset: 0. 0=Disable RtcClk output.
10	<b>RtcClkDrive.</b> Read-write. Cold reset: 1. 0=HIGHDRIVE is tied low for RtcClkOut pad. 1=HIGH-DRIVE is tied high for RtcClkOut pad.
9:5	Reserved.
4	<b>RtcProtectC0_CF.</b> Write-once. Cold reset: 0. 1=RTC RAM index CFh:C0h are locked from read/write.
3	<b>RtcProtectD0_DF.</b> Write-once. Cold reset: 0. 1=RTC RAM index DFh:D0h are locked from read/write.
2	<b>RtcProtectE0_EF.</b> Write-once. Cold reset: 0. 1=RTC RAM index EFh:E0h are locked from read/write.
1	<b>RtcProtectF0_FF.</b> Write-once. Cold reset: 0. 1=RTC RAM index FFh:F0h are locked from read/write.
0	<b>RtcProtect38_3F.</b> Write-once. Cold reset: 0. 1=RTC RAM index 3Fh:38h are locked from read/write.

### PMx58 VRT\_T1

Bits	Description
7:0	<b>VRT_T1.</b> Read-write. Cold reset: 1. This field specifies the time of VRT_Enable being high for RTC battery monitor circuit in milliseconds. To conserve power, the RTC battery is sampled periodically for checking its state of health. VRT_T1 and <b>PMx59[VRT_T2]</b> make up the interval of the checking. When VRT_Enable is high, the battery is being sampled. When VRT_Enable is low, the battery is not being sampled.

**PMx59 VRT\_T2**

Bits	Description
7:0	<b>VRT_T2</b> . Read-write. Cold reset: FFh. This field specifies the time of VRT_Enable being low for the RTC battery monitor circuit in 4 ms increments. See <a href="#">PMx58 [VRT_T1]</a> for detailed description.

**PMx5B RTC Shadow**

Bits	Description
7:4	Reserved.
3:0	<b>PwrFailShadow</b> . Read-write. Cold reset: 0. BIOS: 0100b.

**PMx5C LLBCntrl**

Bits	Description
7:3	Reserved.
2	<b>AllowWakeS3En</b> . Read-write. Cold reset: 0. 1=Allow LLB# as wake event in S3. Note: LLB (Low low battery) event should trigger a SCI if system is in S0. It should block wake-up if the system is in S-States. To meet that required behavior, SW should use the following setting: LLB_En (PMxC8[10]) = 1 BlockWakeEn = 0 UseAsWakeEn = 1 AllowWakeS3En = 0 or 1 (don't care).
1	<b>UseAsWakeEn</b> . Read-write. Cold reset: 0. 1=Treat LLB# as wake event.
0	<b>BlockWakeEn</b> . Read-write. Cold reset: 0. 1=Block wake event if LLB# is asserted; if (UseAsWakeEn==1) && (AllowWakeS3En==1), LLB# and other wake events can wake the system up from S3.

**PMx5E RTC ExtIndex**

Bits	Description
7:0	<b>Index</b> . Read-write. Cold reset: X. Specifies the offset of the RTC Extended Register to be read/written from <a href="#">PMx5F</a> .

**PMx5F RTC ExtData**

Bits	Description
7:0	<b>Data</b> . Read-write. Cold reset: X. Specifies the read data or write data of the RTC Extended Register.

**PMx5F\_x00 RTCEXT DltSavEnable**

Bits	Description
------	-------------

7:1	Reserved.
0	<b>DltSavEnable.</b> Read-write. Reset: 0. 1=Enable RTC daylight saving feature.

**PMx5F\_x01 RTCEXT SprFwdCtrl**

Bits	Description
7	Reserved.
6	<b>SprFwdWeek.</b> Read-write. Reset: 0. This specifies which Sunday morning to do the “spring forward”. Spring forward is usually at the 1st Sunday of April in United States and last Sunday of March in Europe. 0=The 1st Sunday of the month. 1=The last Sunday of the month.
5:0	<b>SprFwdHour.</b> Read-write. Reset: 0. This Binary-Coded Decimal (BCD) value specifies which hour (24 hour mode) to do the “spring forward”. Spring forward is usually 2 am in United States and 1 am in Europe. 0=2 am. 02h=2 am.

**PMx5F\_x02 RTCEXT SprFwdMonth**

Bits	Description
7:5	Reserved.
4:0	<b>SprFwdMonth.</b> Read-write. Reset: 0. This Binary-Coded Decimal (BCD) value determines which month to do the “spring forward”. Spring forward is usually at April in United States and March in Europe. 0=April. 04h=April.

**PMx5F\_x03 RTCEXT FallBackCtrl**

Bits	Description
7	Reserved.
6	<b>FallBackWeek.</b> Read-write. Reset: 0. This value specifies which Sunday morning to do the “fall back”. Fall back is usually at the last Sunday of October in both United States and Europe. 0=The last week of the month. 1=The first week of the month.
5:0	<b>FallBackHour.</b> Read-write. Reset: 0. This Binary-Coded Decimal (BCD) value specifies which hour (24 hour mode) to do the “fall back”. Fall back is usually 2 am in United States and 1 am in Europe. 0=2 am. 02h=2 am.

**PMx5F\_x04 RTCEXT FallBackMonth**

Bits	Description
7:5	Reserved.
4:0	<b>FallBackMonth.</b> Read-write. Reset: 0. This Binary-Coded Decimal (BCD) value specifies which month to “fall back”. Fall back is usually at October in both United States and Europe. 0=October. 10h=October.

**PMx5F\_x10 RTCEXT WeekTimerControl**

The 16-bit Week Timer is a battery-powered down counter timer that supports 1ms, 1 second, and 1 minute resolution and auto reloads when the timer reaches 0. The WEEK\_ALARM interrupt is asserted when the timer reaches 0 and stays asserted until the timer is disabled.

Bits	Description												
7:3	Reserved.												
2:1	<p><b>Resolution.</b> Read-write. Reset: 0. This field specifies the resolution of the Week Timer counter. Before programming this bit, software should program Enable to 0 to disable the Week Timer.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>1 minute</td> <td>10b</td> <td>1 ms</td> </tr> <tr> <td>01b</td> <td>1 second</td> <td>11b</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Definition	Bits	Definition	00b	1 minute	10b	1 ms	01b	1 second	11b	Reserved
Bits	Definition	Bits	Definition										
00b	1 minute	10b	1 ms										
01b	1 second	11b	Reserved										
0	<b>Enable.</b> Read-write. Reset: 0. 0=Disable Week Timer. 1=Enable Week Timer.												

**PMx5F\_x11 RTCEXT WeekTimerReloadLow**

Bits	Description
7:0	<b>WeekTimerReloadLow.</b> Read-write. Reset: 0. This field is used to program the lower 8 bits of the 16-bit WeekTimerReload register. Writing the WeekTimerReloadLow register causes the 16-bit WeekTimerReload to be written into the Week Timer. Software should program PMx5F_x10[Enable] = 0 before writing to this register.

**PMx5F\_x12 RTCEXT WeekTimerReloadHigh**

Bits	Description
7:0	<b>WeekTimerReloadHigh.</b> Read-write. Reset: 0. This field is used to program the upper 8 bits of the 16-bit WeekTimerReload register. This field should be programmed before programming PMx5F_x11 [RTCEXT WeekTimerReloadLow]. Software should program PMx5F_x10[Enable] to 0 before writing to this register.

**PMx5F\_x13 RTCEXT WeekTimerDataLow**

Bits	Description
7:0	<b>WeekTimerDataLow.</b> Read-only. Reset: 0. This field is used to read the current state of the 16-bit Week Timer. Reading from the WeekTimerDataLow register returns the lower 8 bits of the 16-bit Week Timer and causes the upper 8 bits to be latched into PMx5F_x14 [RTCEXT WeekTimerDataHigh].

**PMx5F\_x14 RTCEXT WeekTimerDataHigh**

Bits	Description
7:0	<b>WeekTimerDataHigh.</b> Read-only. Reset: 0. This field is used to read the current state of the 16-bit Week Timer. Reading from the WeekTimerDataHigh register returns the upper 8 bits of the 16-bit Week Timer latched by a previous read from PMx5F_x11 [RTCEXT WeekTimerReloadLow].

**PMx60 AcpiPm1EvtBlk**

Bits	Description
15:2	<b>AcpiPm1EvtBlk</b> . Read-write. Cold reset: 0. Specifies the 16-bit IO range base address[15:2] of the ACPI power management event block defined in 3.25.15.1 [AcpiPmEvtBlk].
1:0	Reserved.

**PMx62 AcpiPm1CntBlk**

Bits	Description
15:1	<b>AcpiPm1CntBlk</b> . Read-write. Cold reset: 0. Specifies the 16-bit IO base address[15:1] of the ACPI power management control block defined in 3.25.15.2 [AcpiPm1CntBlk].
0	Reserved.

**PMx64 AcpiPmTmrBlk**

Bits	Description
15:1	<b>AcpiPmTmrBlk</b> . Read-write. Cold reset: 0. Specifies the 16-bit IO base address[15:1] of the ACPI power management timer block defined in 3.25.15.4 [AcpiPmTmrBlk].
0	Reserved.

**PMx66 CpuCntBlk**

Bits	Description
15:3	<b>CpuControl</b> . Read-write. Cold reset: 0. Specifies the 16-bit IO base address[15:3] of the ACPI power management CPU control block defined in 3.25.15.5 [CpuCntBlk].
2:0	Reserved.

**PMx68 AcpiGpe0Blk**

Bits	Description
15:2	<b>AcpiGpe0Blk</b> . Read-write. Cold reset: 0. Specifies the 16-bit IO base address[15:2] of the ACPI power management general purpose event block defined in 3.25.15.6 [AcpiGpe0Blk].
1:0	Reserved.

**PMx6A AcpiSmiCmd**

Bits	Description
15:0	<b>AcpiSmiCmd</b> . Read-write. Cold reset: 0. These bits define the 16-bit IO base address[15:0] of the ACPI SMI command block defined in 3.25.15.7 [SmiCmdBlk]. The address is required to be WORD-aligned (Addr[0]=0).



**PMx6E AcpiPm2CntBlk**

Bits	Description
15:0	<b>AcpiPm2CntBlk.</b> Read-write. Cold reset: 0. These bits define the 16-bit IO base address[15:0] of the ACPI power management additional control block defined in 3.25.15.3 [AcpiPm2CntBlk].

**PMx74 AcpiConfig**

Bits	Description
31:30	Reserved.
29	<b>RtcWakeAlarm.</b> Read-write. Cold reset: 1. BIOS: 0. 0=Use IO073_x0C[IRQF] as one of the system wake-up event if IRQF is enabled. 1=Use IO073_x0C[AF] as one of the system wake-up event if AF is enabled.
28	<b>PcieGeventMap.</b> Read-write. Cold reset: 0. 1=Route PME message from NB to GEvent 24, hot plug message from APU to GEvent 7.
27	<b>WakePinAsGevent.</b> Read-write. Cold reset: 0. 1=Treat Wake# pin as GEvent input.
26	Reserved.
25	<b>PcieWakMask.</b> Read-write. Cold reset: 0. 1=Disable the PCIE_WAK_STS and PCIE_WAK_DIS function defined in AcpiPmEvtBlkx00[PCIExpWakeStatus] and AcpiPmEvtBlkx02[PCIExpWakeDis].
24	<b>PcieNative.</b> Read-write. Cold reset: 0. 1=Block PCIe GPP PME message and hot plug message from generating SCI.
23:8	Reserved.
7	<b>BiosRls.</b> RAZ; write-1-only. Cold reset: 0. Writing 1 to this bit generates SMI, NMI or IRQ13 depending on SMIXB0[Smicontrol73].
6	<b>MaskArbDis.</b> Read-write. Cold reset: 1.
5	Reserved.
4	<b>TmrEnEn.</b> Read-write. Cold reset: 0. 1=Enable the TMR_EN function defined in AcpiPmEvtBlkx02[TmrEn].
3	Reserved.
2	<b>RtcEnEn.</b> Read-write. Cold reset: 0. 1=Enable the RTC_EN function defined in AcpiPmEvtBlkx02[RtcEn].
1	<b>GblEnEn.</b> Read-write. Cold reset: 0. 1=Enable the GBL_EN function defined in AcpiPmEvtBlkx02[GblEn].
0	<b>DecEnAcpi.</b> Read-write. Cold reset: 0. 1=Enable decoding of the standard ACPI registers.

**PMx78 WakeIoAddr**

Bits	Description
15:0	<b>WakeIoBaseAddress.</b> Read-write. Reset: FFFFh.

**PMx7A HaltCountEn**

Bits	Description
15	<b>CountHaltMsgEn.</b> Read-write. Reset: 0. 1=FCH keeps track of the APU state by counting HALT entering and exit messages; When the number of net HALT enter messages matches with NumOfHalt, FCH initiates C1e.
14:4	Reserved.
3:0	<b>NumOfHalt.</b> Read-write. Reset: 0. Defines the number of HALT messages to track before FCH should initiate C1e sequence.

**PMx7C C1eWrPortAdr**

Bits	Description
15:0	<b>C1eWrPortAdr.</b> Read-write. Reset: FFFFh.

**PMx7E CStateEn**

Bits	Description
15	<b>MaskIntrEn.</b> Read-write. Reset: 1. 1=APIC interrupts are deferred until the first ACPI access after the system resumes from S state; In addition, A20M, IGNNE, INTR, NMI, INIT messages are deferred the same way, but SMI is not be deferred. This is mainly to guard against unexpected interrupt being sent to OS during S1 resume.
14	<b>MaskCStateSys.</b> Read-write. Reset: 0. 1=Skip C state if there is a pending interrupt such as SMI, NMI, and INIT request.
13:7	Reserved.
6	<b>CPopUpEn.</b> Read-write. Reset: 1. 1=Enable pop up capability, which means APU goes to C2 if there is a traffic and comes back to C3 after being idle for a while. This bit should be set to 0 when AltVid is not enabled.
5	<b>C1eToC3En.</b> Read-write. Reset: 1. 1=Put APU into C3 state in C1e state.
4	<b>C1eToC2En.</b> Read-write. Reset: 0. 1=Put APU into C2 state in C1e state.
3	<b>C2EnhanceEn.</b> Read-write. Reset: 0. 1=Enable C2 enhancement.
2	<b>C2ToC3Enable.</b> Read-write. Reset: 0. 1=Put APU into C3 even in the case of IO reads to <a href="#">CpuCntBlkx04 [PLvl2]</a> .
1:0	Reserved.

**PMx80 Break Event**

Bits	Description
31	Reserved.
30:24	<b>AutoStutterLimit.</b> Read-write. Cold reset: 0. This specifies the limit for the AutoStutterTimer. Time unit is defined by [AutoStutterTimeSel]. See [AutoStutterTimerEn] for detailed description on AutoStutterTimer.

23:22	Reserved.
21	<b>UsbPeriodicalSetBmSts.</b> Read-write. Cold reset: 1. 1=Treat USB isochronous traffic as source of DMA traffic; this is to serve as a quicker way to bring the link between FCH and Fusion back to the active state.
20:19	Reserved.
18	<b>Usb20SetBmSts.</b> Read-write. Cold reset: 1. 1=Treat USB 2.0 traffic as source of DMA traffic; this is to serve as a quicker way to bring the link between FCH and Fusion back to the active state.
17	<b>Usb11SetBmSts.</b> Read-write. Cold reset: 1. 1=Treat USB 1.1 traffic as source of DMA traffic; this is to serve as a quicker way to bring the link between FCH and Fusion back to the active state.
16	<b>Usb11BmStsEn.</b> Read-write. Cold reset: 0. 1=Any USB 1.1 activity sets <a href="#">AcpiPmEvtBlkx00[BmStatus]</a> to 1.
15:14	Reserved.
13	<b>ServerCEn.</b> Read-write. Cold reset: 0. 1=FCH monitors the HALT messages coming from APU(s). When all APUs are in HALT state, FCH automatically transitions to C3/C1e state. If any APU exits from HALT state, FCH will exit out from C3/C1e state as well. This is used in conjunction with <a href="#">PMx7A</a> .
12	<b>AutoStutterTimeSel.</b> Read-write. Cold reset: 0. This bit selects the time increment used by the AutoStutterTimer. 0=2 us increment. 1=1 ms increment.
11	<b>AutoStutterTimerEn.</b> Read-write. Cold reset: 0. 1=Enable an AutoStutterTimer that automatically counts whenever LDTSTOP is asserted during C3/C1e state. When the timer reaches the threshold defined by AutoStutterLimit, it will stutter the C state machine. This feature is specifically designed to periodically reconnect the HT link in the case of a long idle time for multi-CPU's.
10	<b>MergeBMReqEn.</b> Read-write. Cold reset: 0. 1=The logic merges BMREQ# and AllowLdtStop together internally.
9	<b>BusReqHoldEn.</b> Read-write. Cold reset: 0. 1=Extend BMREQ# until LDTSTP# is asserted.
8	<b>BmReqPopUpEn.</b> Read-write. Cold reset: 0. 1=Allow PopUp if BMREQ# is toggled when PopUp function is enabled.
7	<b>BmReqEn.</b> Read-write. Cold reset: 0. 1=Treat BMREQ# as one source of Break Event.
6:5	Reserved.
4	<b>EnableBreak.</b> Read-write. Cold reset: 0. 1=Skip the C state transition if there is break event when entering C state.
3	Reserved.
2	<b>AutoClrBmSts.</b> Read-write. Cold reset: 0. It is used for PopUp and C1e function. 1=Automatically clear <a href="#">AcpiPmEvtBlkx00[BmStatus]</a> before entering C1e state.
1	<b>AutoBmRld.</b> Read-write. Cold reset: 0. 1=Generate an internal <a href="#">AcpiPmEvtBlkx00[BmStatus]</a> enable bit (that is similar to <a href="#">AcpiPm1CntBlkx00[BmRld]</a> ) upon entry to C1e. Depending on the configuration of other bits, bus master activity or IDLE_EXIT# pin could cause FCH to break out from C1e.
0	<b>BmStsRdMask.</b> Read-write. Cold reset: 0. 0= <a href="#">AcpiPmEvtBlkx00[BmStatus]</a> is set to 1 if there is any DMA traffic in FCH. 1=Make <a href="#">AcpiPmEvtBlkx00[BmStatus]</a> always return 0 except for USB 1.1 traffic.

**PMx88 CStateControl**

Bits	Description
31:14	Reserved.
13	<b>LdtStpCmd.</b> Read-write. Reset: 0. Programming it to 1 from 0 forces FCH to toggle LDTSTP#.
12	<b>StutterMode.</b> Read-write. Reset: 0. 1=Enable stutter mode.
11:6	Reserved.
5	<b>SlpEn.</b> Read-write. Reset: 0. 1=Enable LDTSTOP# as an output.
4	<b>CcEn.</b> Read-write. Reset: 0. 1=Enable C State. This bit must be set in order to exercise C state.
3	Reserved.
2	<b>DlySlpEn.</b> Read-write. Reset: 0. 1=Delay recognition of STPGNT# until there is no pending read in AB.
1	Reserved.
0	<b>WaitStpGntEnB.</b> Read-write. Reset: 0. 1=Wait for STPGNT# in ACPI S state.

**PMx8E PopUpEndTime**

Bits	Description
7:0	<b>PopUpEndTime.</b> Read-write. Cold reset: 10h.

**PMx94 CStateTiming0**

Bits	Description												
31:26	Reserved.												
25:24	<b>LdtEndTime.</b> Read-write. Cold reset: 0. Specifies the delay from deassertion of LDTSTP# till the deassertion of STPCLK. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Bits</th> <th>Definition</th> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>0 us</td> <td>10b</td> <td>32 us</td> </tr> <tr> <td>01b</td> <td>1 us</td> <td>11b</td> <td>64 us</td> </tr> </tbody> </table>	Bits	Definition	Bits	Definition	00b	0 us	10b	32 us	01b	1 us	11b	64 us
Bits	Definition	Bits	Definition										
00b	0 us	10b	32 us										
01b	1 us	11b	64 us										
23:16	<b>LdtStartTime.</b> Read-write. Cold reset: 10h. Specifies the LDTSTP# deassertion time in 1 us increments in C state.												
15:8	<b>SLdtStartTime.</b> Read-write. Cold reset: 0. Specifies the delay between LPC_PD# assertion and LDTSTP# assertion when the system enters ACPI S states. The time is in 1 us increments, with 1 us uncertainty.												
7:0	<b>StutterTime.</b> Read-write. Cold reset: 1. Specifies the LDTSTP# duration in 1 us increments. This is basically the minimum LDTSTP# assertion time.												

**PMx98 CStateTiming1**

Bits	Description
31	Reserved.

30:28	<b>VidFidTime.</b> Read-write. Cold reset: 1. Specifies the VID/FID LDTSTP# duration.																				
	<table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>0 us</td> <td>100b</td> <td>16 us</td> </tr> <tr> <td>001b</td> <td>2 us</td> <td>101b</td> <td>32 us</td> </tr> <tr> <td>010b</td> <td>4 us</td> <td>110b</td> <td>64 us</td> </tr> <tr> <td>011b</td> <td>8 us</td> <td>111b</td> <td>128 us</td> </tr> </tbody> </table>	Bits	Definition	Bits	Definition	000b	0 us	100b	16 us	001b	2 us	101b	32 us	010b	4 us	110b	64 us	011b	8 us	111b	128 us
Bits	Definition	Bits	Definition																		
000b	0 us	100b	16 us																		
001b	2 us	101b	32 us																		
010b	4 us	110b	64 us																		
011b	8 us	111b	128 us																		
27:16	Reserved.																				
15:8	<b>FirstLdtStartTime.</b> Read-write. Cold reset: 10h. Specifies the very first LDTSTP# assertion delay in 1 us increments from reception of STPGNT.																				
7:0	<b>StpClkDlyTime.</b> Read-write. Cold reset: 0. Specifies the additional STPCLK# deassertion delay in number of oscillator clocks for S1 resume.																				

**PMx9C C2Count**

Bits	Description
7:0	<b>C2Count.</b> Read-only; updated-by-hardware. Reset: 0. Specifies the amount of time the APU spends in STPGNT state (but LDTSTOP# is not asserted) during C1e. Each increment is approximately 0.39% (1/256). This field is updated every second.

**PMx9D C3Count**

Bits	Description
7:0	<b>C3Count.</b> Read-only; updated-by-hardware. Reset: 0. Specifies the amount of time LDTSTOP# is asserted during C1e. Each increment is approximately 0.39% (1/256). This field is updated every second.

**PMxA0 MessageCState**

Bits	Description
31	<b>CheckCoreIdDis.</b> Read-write. Reset: 0. 1=Ignore core ID check if (MultiCoreEn == 1). 0=Enable core ID check if (MultiCoreEn == 1).
30:24	<b>ExtendValue.</b> Read-write. Reset: 0. Specifies the timer value to be used with ExtendEnable. The value is the number of 66 MHz clocks.
23	<b>MultiCoreEn.</b> Read-write. Reset: 0. 1=C state control logic inside FCH assumes multi-core configuration; <b>PMx7A</b> [NumOfHalt] should be programmed accordingly; FCH keeps track of APU C state by monitoring each core's C state message instead of package state.
22	<b>FusionExitCMsgDis.</b> Read-write. Reset: 0. 1=ExitC Message will be blocked.
21	<b>FusionAllowCMsgDis.</b> Read-write. Reset: 0. 1=AllowC Message will be blocked.

20:16	<p><b>TmrSelOverride.</b> Read-write. Reset: 0. These bits are used with TimerTickChgMsgEn. In case that FCH auto-timer detection logic doesn't function properly, these bits can be used to override the logic and force the logic to monitor the specific timer.</p> <table border="0"> <thead> <tr> <th><u>Bit</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>[4]</td> <td>1=Force the logic to monitor HPET timer 2 if HPET is selected</td> </tr> <tr> <td>[3]</td> <td>1=Force the logic to monitor HPET timer 1 if HPET is selected</td> </tr> <tr> <td>[2]</td> <td>1=Force the logic to monitor HPET timer 0 if HPET is selected</td> </tr> <tr> <td>[1]</td> <td>1=Use RTC</td> </tr> <tr> <td>[0]</td> <td>1=Use HPET</td> </tr> </tbody> </table>	<u>Bit</u>	<u>Definition</u>	[4]	1=Force the logic to monitor HPET timer 2 if HPET is selected	[3]	1=Force the logic to monitor HPET timer 1 if HPET is selected	[2]	1=Force the logic to monitor HPET timer 0 if HPET is selected	[1]	1=Use RTC	[0]	1=Use HPET
<u>Bit</u>	<u>Definition</u>												
[4]	1=Force the logic to monitor HPET timer 2 if HPET is selected												
[3]	1=Force the logic to monitor HPET timer 1 if HPET is selected												
[2]	1=Force the logic to monitor HPET timer 0 if HPET is selected												
[1]	1=Use RTC												
[0]	1=Use HPET												
15:14	<p><b>UsbEhciModeFusion[1:0].</b> Read-write. Reset: 0. These bits are used with Fusion CPU C state.</p> <table border="0"> <thead> <tr> <th><u>Bit</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>[1]</td> <td>1=FCH causes CPU to break out from C state when USB EHCI has pending traffic.</td> </tr> <tr> <td>[0]</td> <td>1=FCH stutters the CPU C state whenever USB EHCI has pending traffic.</td> </tr> </tbody> </table>	<u>Bit</u>	<u>Definition</u>	[1]	1=FCH causes CPU to break out from C state when USB EHCI has pending traffic.	[0]	1=FCH stutters the CPU C state whenever USB EHCI has pending traffic.						
<u>Bit</u>	<u>Definition</u>												
[1]	1=FCH causes CPU to break out from C state when USB EHCI has pending traffic.												
[0]	1=FCH stutters the CPU C state whenever USB EHCI has pending traffic.												
13:12	<p><b>UsbOhciModeFusion[1:0].</b> Read-write. Reset: 0. These bits are used with Fusion CPU C state.</p> <table border="0"> <thead> <tr> <th><u>Bit</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>[1]</td> <td>1=FCH causes CPU to break out from C state when USB OHCI has pending traffic.</td> </tr> <tr> <td>[0]</td> <td>1=FCH stutters the CPU C state whenever USB OHCI has pending traffic.</td> </tr> </tbody> </table>	<u>Bit</u>	<u>Definition</u>	[1]	1=FCH causes CPU to break out from C state when USB OHCI has pending traffic.	[0]	1=FCH stutters the CPU C state whenever USB OHCI has pending traffic.						
<u>Bit</u>	<u>Definition</u>												
[1]	1=FCH causes CPU to break out from C state when USB OHCI has pending traffic.												
[0]	1=FCH stutters the CPU C state whenever USB OHCI has pending traffic.												
11	<b>ExtendEnable.</b> Read-write. Reset: 0. 1=Enable FCH to start a timer whenever the C state is exited, in order to prevent CPU from going back to C state before the timer expires. See ExtendValue for the timer.												
10	<b>ClkIntrTagEn.</b> Read-write. Reset: 0. 1=Enable FCH to mark the periodic timer interrupt.												
9	<b>PopUpByEc.</b> Read-write. Reset: 0. A static bit that can be written by IMC to cause FCH to pop up C state message to CPU.												
8	<b>WakeByEc.</b> Read-write. Reset: 0. A static bit that can be written by IMC to cause FCH to generate C state exit message to CPU.												
7	<b>SelfExitEnable.</b> Read-write. Reset: 0. 1=FCH exits to C0 state from non-C0 state when there is a break event, and exits to C1 state from non-C0 state when there is traffic.												
6	<b>BatteryModeEn.</b> Read-write. Reset: 0. 1=A change in power mode (battery vs AC) causes FCH to tell CPU to exit from C state.												
5	Reserved.												
4	<b>FusionSerrEn.</b> Read-write. Reset: 0. 1=FCH C state coordination logic causes CPU to exit from C state when there is a system error within the FCH.												
3	<b>FusionPerrEn.</b> Read-write. Reset: 0. 1=FCH C state coordination logic causes CPU to exit from C state when there is a parity error within the FCH.												
2	<b>FusionCEnable.</b> Read-write. Reset: 1=Enable the FCH Fusion C state coordination logic. Fusion CPU contains the actual C state logic and FCH contains the coordination logic which sends handshake message to CPU to help it to decide which C state to go into.												
1	<b>TimerTickChgMsgEn.</b> Read-write. Reset: 0. 1=FCH sends a message to APU indicating the latest periodic timer interval. FCH automatically determines which timer (PIT, RTC, or HPET) is being used. See TmrSelOverride.												
0	<b>BattModeChgMsgEn.</b> Read-write. Reset: 0. 1=FCH automatically sends a message to CPU indicating the power mode (AC vs battery); In addition, every time it is changed, FCH generates a message to indicate the update.												

**PMxA4 Traffic Monitor and Status**Address: **PMxA4**[PmioA4Sel].

Bits	Description
31	<b>PmioA4Sel.</b> Read-write. Reset: 0. 0=This register is the TrafficMonitorEn register. 1=This register is selected to be TrafficStatus register. In both cases, this bit has the same definition.
30:0	<b>TrafficData.</b>

**PMxA4\_x0 TrafficMonitorEn**

Bits	Description										
31	<b>PmioA4Sel.</b> Read-write. Reset: 0. 0=This register is the TrafficMonitorEn register. 1=This register is selected to be TrafficStatus register. In both cases, this bit has the same definition.										
30	<b>IntrSciSts.</b> Read-only. Reset: 0. This bit can be cleared by programming InterruptSciEn to 0. 0=SCI event from interrupt monitoring doesn't happen. 1=SCI event from interrupt monitoring has been fired. See InterruptSciEn for details on interrupt monitoring.										
29	<b>TrafficSciSts.</b> Read-only. Reset: 0. This bit can be cleared by programming TrafficSciEn to 0. 0=SCI event from traffic monitoring doesn't happen. 1=SCI event from traffic monitoring has been fired. See TrafficSciEn for details on traffic monitoring.										
28	<b>ModFusionCHandShakeEn.</b> Read-write. Reset: 0. 0=Fusion C-state logic uses a 3-way handshake protocol: CPU -> C1 entry message FCH -> C-state allow message CPU -> C-state taken message 1=Fusion C-state logic uses a 2-way handshake protocol: CPU -> C1 entry message FCH -> C-state allow message From there, FCH assumes CPU will take the highest C state indicated by the C-state allow message.										
27	<b>IntrLess.</b> Read-write. Reset: 0. 0=Monitor whether the number of interrupts is more than <b>PMxAA</b> [IntrTimeLimit]. 1=Monitor whether the number of interrupts is less than <b>PMxAA</b> [IntrTimeLimit].										
26	<b>TrafficLess.</b> Read-write. Reset: 0. 0=Monitor whether the idle number is less than <b>PMxA8</b> [IdleTimeLimit]. 1=Monitor whether the idle number is more than <b>PMxA8</b> [IdleTimeLimit].										
25:24	<b>PerfMonPeriodSel.</b> Read-write. Reset: 0. Specifies the traffic/interrupt monitoring period. It is used when [PerfMonEn] == 1. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>15 ns between each count and the monitored interval is ~1 ms.</td> </tr> <tr> <td>01b</td> <td>240 ns between each count and the monitored interval is 15.67 ms.</td> </tr> <tr> <td>10b</td> <td>1.92 us between each count and the monitored interval is 125.3 ms.</td> </tr> <tr> <td>11b</td> <td>15.36 us between each count and the monitored interval is ~1 second.</td> </tr> </tbody> </table>	Bits	Definition	00b	15 ns between each count and the monitored interval is ~1 ms.	01b	240 ns between each count and the monitored interval is 15.67 ms.	10b	1.92 us between each count and the monitored interval is 125.3 ms.	11b	15.36 us between each count and the monitored interval is ~1 second.
Bits	Definition										
00b	15 ns between each count and the monitored interval is ~1 ms.										
01b	240 ns between each count and the monitored interval is 15.67 ms.										
10b	1.92 us between each count and the monitored interval is 125.3 ms.										
11b	15.36 us between each count and the monitored interval is ~1 second.										
23:15	Reserved.										
14	<b>MaskIntrToCState.</b> Read-write. Reset: 0. 1=Interrupts are masked off internally and do not cause FCH to generate Exit message to Fusion CPU.										
13	Reserved.										

12	<b>CheckFc.</b> Read-write. Reset: 0. This bit is used with TrafficSciEn. 1=Enable FCH to monitor FC (NAND flash) traffic.
11	Reserved.
10	<b>CheckLpc.</b> Read-write. Reset: 0. This bit is used with TrafficSciEn. 1=Enable FCH to monitor LPC traffic.
9	<b>CheckAz.</b> Read-write. Reset: 0. This bit is used with TrafficSciEn. 1=Enable FCH to monitor HD audio traffic.
8	Reserved.
7	<b>CheckUsb.</b> Read-write. Reset: 0. This bit is used with TrafficSciEn. 1=Enable FCH to monitor USB traffic.
6	<b>CheckSata.</b> Read-write. Reset: 0. This bit is used with TrafficSciEn. 1=Enable FCH to monitor SATA traffic.
5	Reserved.
4	<b>CheckC3.</b> Read-write. Reset: 0. This bit is used with TrafficSciEn. 1=Enable FCH to only consider the system as in idle state if CPU is in C state.
3	<b>CheckInterrupt.</b> Read-write. Reset: 0. This bit is used with InterruptSciEn. 1=Enable FCH to monitor interrupts.
2	<b>InterruptSciEn.</b> Read-write. Reset: 0. 1=FCH monitors the number of interrupts within the monitored interval; If the number of interrupts is less than the number defined in <a href="#">PMxAA[IntrTimeLimit]</a> , FCH generates an SCI.
1	<b>TrafficSciEn.</b> Read-write. Reset: 0. 1=FCH generates an SCI when the amount of traffic is less than <a href="#">PMxA8[IdleTimeLimit]</a> within the monitored interval.
0	<b>PerfMonEn.</b> Read-write. Reset: 0. 1=FCH monitors the amount of DMA traffic enabled by bits [12:6] and the number of interrupts within the monitored interval defined by PerfMonPeriodSel.

**PMxA4\_x1 TrafficStatus**

Bits	Description												
31	<b>PmioA4Sel.</b> Read-write. Reset: 0. 0=This register is the TrafficMonitorEn register. 1=This register is selected to be TrafficStatus register. In both cases, this bit has the same definition.												
30:24	Reserved.												
23:22	<b>FusionCState3.</b> Read-write. Reset: 0. Status of fusion C state monitor for core pair 3. See FusionC-State.												
21:20	<b>FusionCState2.</b> Read-write. Reset: 0. Status of fusion C state monitor for core pair 2. See FusionC-State.												
19:18	<b>FusionCState1.</b> Read-write. Reset: 0. Status of fusion C state monitor for core pair 1. See FusionC-State.												
17:16	<b>FusionCState0.</b> Read-write. Reset: 0. Status of fusion C state monitor for core pair 0. See FusionC-State.												
15:14	<b>FusionCState.</b> Read-write. Reset: 0. Status of fusion C state monitor. <table border="0" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;"><u>Bits</u></th> <th style="text-align: left;"><u>Definition</u></th> <th style="text-align: left;"><u>Bits</u></th> <th style="text-align: left;"><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Fusion Idle</td> <td>10b</td> <td>Fusion C4</td> </tr> <tr> <td>01b</td> <td>Fusion C5/6</td> <td>11b</td> <td>Fusion C3</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	<u>Bits</u>	<u>Definition</u>	00b	Fusion Idle	10b	Fusion C4	01b	Fusion C5/6	11b	Fusion C3
<u>Bits</u>	<u>Definition</u>	<u>Bits</u>	<u>Definition</u>										
00b	Fusion Idle	10b	Fusion C4										
01b	Fusion C5/6	11b	Fusion C3										



13	Reserved.
12	<b>FcTrafficStatus.</b> Read-only; updated-by-hardware. Reset: X. Indicates whether Flash Controller (NAND flash) is active. 0=Inactive. 1=Active.
11	Reserved.
10	<b>LpcTrafficStatus.</b> Read-only; updated-by-hardware. Reset: X. Indicates whether LPC is active. 0=Inactive. 1=Active.
9	<b>AzTrafficStatus.</b> Read-only; updated-by-hardware. Reset: X. Indicates whether HD audio is active. 0=Inactive. 1=Active.
8	Reserved.
7	<b>UsbTrafficStatus.</b> Read-only; updated-by-hardware. Reset: X. Indicates whether USB is active. 0=Inactive. 1=Active.
6	<b>SataTrafficStatus.</b> Read-only; updated-by-hardware. Reset: X. Indicates whether SATA is active. 0=Inactive. 1=Active.
5:0	Reserved.

### PMxA8 TrafficMonitorIdleTime

---

Bits	Description
15:0	<b>IdleTimeLimit.</b> Read-write. Reset: 0. This field is used with <a href="#">PMxA4_x0[PerfMonEn]</a> . It specifies the amount of DMA traffic limit that causes FCH to generate an SCI. Time granularity is defined by <a href="#">PMxA4_x0[PerfMonPeriodSel]</a> .

### PMxAA TrafficMonitorIntTime

---

Bits	Description
15:0	<b>IntrTimeLimit.</b> Read-write. Reset: 0. This field is used with <a href="#">PMxA4_x0[PerfMonEn]</a> . It specifies the amount of interrupt limit that causes FCH to generate an SCI. Time granularity is defined by <a href="#">PMxA4_x0[PerfMonPeriodSel]</a> .

### PMxAC TrafficMonitorTrafficCount

---

Bits	Description
15:0	<b>TrafficCount.</b> Read-only; updated-by-hardware. Reset: 0. Specifies the actual recorded value of the combined DMA traffic during the monitored period.

### PMxAE TrafficMonitorIntrCount

---

Bits	Description
15:0	<b>IntrCount.</b> Read-only; updated-by-hardware. Reset: 0. Specifies the actual recorded value of the number of interrupts during the monitored period.

**PMxB0 DeferTimeTick**

Bits	Description																				
31:12	Reserved.																				
11	Reserved.																				
10:8	<b>DeferTimerTickValue.</b> Read-write. Reset: 0. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>No skipping</td> <td>100b</td> <td>Skip 4 timer ticks</td> </tr> <tr> <td>001b</td> <td>Skip 1 timer tick</td> <td>101b</td> <td>Skip 5 timer ticks</td> </tr> <tr> <td>010b</td> <td>Skip 2 timer ticks</td> <td>110b</td> <td>Skip 6 timer ticks</td> </tr> <tr> <td>011b</td> <td>Skip 3 timer ticks</td> <td>111b</td> <td>Skip 7 timer ticks</td> </tr> </tbody> </table>	Bits	Definition	Bits	Definition	000b	No skipping	100b	Skip 4 timer ticks	001b	Skip 1 timer tick	101b	Skip 5 timer ticks	010b	Skip 2 timer ticks	110b	Skip 6 timer ticks	011b	Skip 3 timer ticks	111b	Skip 7 timer ticks
Bits	Definition	Bits	Definition																		
000b	No skipping	100b	Skip 4 timer ticks																		
001b	Skip 1 timer tick	101b	Skip 5 timer ticks																		
010b	Skip 2 timer ticks	110b	Skip 6 timer ticks																		
011b	Skip 3 timer ticks	111b	Skip 7 timer ticks																		
7:2	Reserved.																				
1	<b>ForceTmrTickEn.</b> Read-write. Reset: 0. 1=If (DeferTimerTickEn == 1) && FCH has skipped a timer tick interrupt, FCH immediately generates the timer tick interrupt upon C state exit.																				
0	<b>DeferTimerTickEn.</b> Read-write. Reset: 0. 1=FCH skips a number of timer tick interrupts based on the value defined in DeterTimeTickValue when CPU is in C state. When CPU is not in C state, FCH does not skip any timer tick interrupt.																				

**PMxB4 Tpreset1b**

Bits	Description																				
31:30	Reserved.																				
29:24	<b>Tpreset1b.</b> Read-write. Cold reset: 5h. This is the timing parameter used for S*->S0 state transition. It specifies the delay between CPU_STP# de-assertion and LPC_PD# de-assertion, in 8 us increment with 8 us uncertainty.																				
23:12	Reserved.																				
11:9	<b>FidVidOption.</b> Read-write. Cold reset: 0. Specifies the additional FID/VID exit delay. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>0 ns</td> <td>100b</td> <td>490 ns</td> </tr> <tr> <td>001b</td> <td>140 ns</td> <td>101b</td> <td>560 ns</td> </tr> <tr> <td>010b</td> <td>210 ns</td> <td>110b</td> <td>350 ns</td> </tr> <tr> <td>011b</td> <td>280 ns</td> <td>111b</td> <td>420 ns</td> </tr> </tbody> </table>	Bits	Definition	Bits	Definition	000b	0 ns	100b	490 ns	001b	140 ns	101b	560 ns	010b	210 ns	110b	350 ns	011b	280 ns	111b	420 ns
Bits	Definition	Bits	Definition																		
000b	0 ns	100b	490 ns																		
001b	140 ns	101b	560 ns																		
010b	210 ns	110b	350 ns																		
011b	280 ns	111b	420 ns																		
8	<b>DelayLdtStp.</b> Read-write. Cold reset: 0. 1=Enable LDTSTP# assertion time.																				
7:1	Reserved.																				
0	<b>FidProtectEn.</b> Read-write. Cold reset: 0. 1=Skip C-state transition when FID/VID messages are received concurrently.																				

**PMxB8 Tpreset2**

Bits	Description										
7:6	<b>ClkGateCntrl.</b> Read-write. Cold reset: 2h. These two bits control whether SMBUS module allows clock gating to the internal 66 MHz core clock. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Disable the clock gating function</td> </tr> <tr> <td>01b</td> <td>Wait 16 clocks before allowing clock gating to the SMBUS module</td> </tr> <tr> <td>10b</td> <td>Wait 64 clocks before allowing clock gating to the SMBUS module</td> </tr> <tr> <td>11b</td> <td>Wait 256 clocks before allowing clock gating to the SMBUS module</td> </tr> </tbody> </table>	Bits	Definition	00b	Disable the clock gating function	01b	Wait 16 clocks before allowing clock gating to the SMBUS module	10b	Wait 64 clocks before allowing clock gating to the SMBUS module	11b	Wait 256 clocks before allowing clock gating to the SMBUS module
Bits	Definition										
00b	Disable the clock gating function										
01b	Wait 16 clocks before allowing clock gating to the SMBUS module										
10b	Wait 64 clocks before allowing clock gating to the SMBUS module										
11b	Wait 256 clocks before allowing clock gating to the SMBUS module										
5:0	<b>Tpreset2.</b> Read-write. Cold reset: 8h. This is the timing parameter used for S*->S0 state transitions. It specifies the LDTSTP# deassertion delay in 8 us increment with 8 us uncertainty.										

**PMxB9 LpcMisc**

Bits	Description
7:4	Reserved.
3	<b>ClkRunDisable.</b> Read-write. Cold reset: 0. Legacy DMA and serial IRQ logic reside in this module and they are running on the 33 MHz LPCCLK. 1=Disable this module's ability to support CLKRUN# function from PCIBridge; In other words, this module prevents PCIBridge from stopping the 33 MHz clock.
2:0	Reserved.

**PMxBA S\_StateControl**

Bits	Description
15	<b>MaskPmeMsgEn.</b> Read-write. Cold reset: 0. 1=If (PmeMsgEn == 1), PmeAck messages are ignored and ACPI S state logic solely uses the timeout mechanism to sequence through the S3 state.
14	<b>WakePinEnable.</b> Read-write. Cold reset: 0. 1=Enable wake up from WAKE# pin.
13	Reserved.
12:4	Reserved.
3	<b>PmeMsgEn.</b> Read-write. Cold reset: 0. 1=Enable PmeTurnOff/PmeMsgAck handshake.
2	Reserved.
1	Reserved.
0	<b>LongSLPS3.</b> Read-write. Cold reset: 0. 1=Extend SLP_S3# assertion to 1 s minimum.

**PMxBC ThrottlingControl**

Bits	Description
15	<b>Therm2SecDelay.</b> Read-write. Cold reset: 0. 1=Enable 2 second delay for thermal clock throttle. This bit affects both hardware and software throttle.

14	<b>NoWaitStpGntEn.</b> Read-write. Cold reset: 0. 0=Wait for STPGNT after asserting STPCLK. 1=Do not wait for STPGNT after asserting STPCLK. This bit affects both hardware and software throttle.																				
13	<b>ThermThrotPeriod.</b> Read-write. Cold reset: 0. Specifies the clock throttle period for hardware thermal throttle. 0=30 us. 1=244 us.																				
12:8	Reserved.																				
7	<b>ThrottleControl[3].</b> Read-write. Cold reset: 0. 1=Enable hardware thermal clock throttle.																				
6:4	<b>ThrottleControl[2:0].</b> Read-write. Cold reset: 0. Specifies the throttle interval for STPCLK de-assertion in hardware thermal clock throttle: <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;"><u>Bits</u></th> <th style="text-align: left;"><u>Definition</u></th> <th style="text-align: left;"><u>Bits</u></th> <th style="text-align: left;"><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>50%</td> <td>100b</td> <td>50%</td> </tr> <tr> <td>001b</td> <td>12.5%</td> <td>101b</td> <td>62.5%</td> </tr> <tr> <td>010b</td> <td>25%</td> <td>110b</td> <td>75%</td> </tr> <tr> <td>011b</td> <td>37.5%</td> <td>111b</td> <td>87.5%</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	<u>Bits</u>	<u>Definition</u>	000b	50%	100b	50%	001b	12.5%	101b	62.5%	010b	25%	110b	75%	011b	37.5%	111b	87.5%
<u>Bits</u>	<u>Definition</u>	<u>Bits</u>	<u>Definition</u>																		
000b	50%	100b	50%																		
001b	12.5%	101b	62.5%																		
010b	25%	110b	75%																		
011b	37.5%	111b	87.5%																		
3:2	Reserved.																				
1:0	<b>AcpiThrotPeriod.</b> Read-write. Cold reset: 0. Specifies the clock throttle period for software thermal throttle. See <a href="#">CpuCntBlkx00 [ClkValue]</a> for software thermal throttle. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;"><u>Bits</u></th> <th style="text-align: left;"><u>Definition</u></th> <th style="text-align: left;"><u>Bits</u></th> <th style="text-align: left;"><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>15 us</td> <td>10b</td> <td>244 us</td> </tr> <tr> <td>01b</td> <td>30 us</td> <td>11b</td> <td>Reserved</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	<u>Bits</u>	<u>Definition</u>	00b	15 us	10b	244 us	01b	30 us	11b	Reserved								
<u>Bits</u>	<u>Definition</u>	<u>Bits</u>	<u>Definition</u>																		
00b	15 us	10b	244 us																		
01b	30 us	11b	Reserved																		

**PMxBE ResetControl1**

**Table 238: BIOS Recommendations for KbRstEn**

Case	Condition	Setting	Comments
1	If the KBRST# / GEVENT1# I/O pin is not connected to system keyboard reset or is configured as GEvent1 function.	0	This bit must be cleared by the platform system BIOS in this case.
2	All other situations other than that specified in Case 1	Do not program.	Value should remain at power-up default setting.

Bits	Description
7	<b>RstToCpuPwrGdEn.</b> Read-write. Cold reset: 0. 1=FCH toggles CpuPwrGd on every reset.
6	<b>HwmResetOption.</b> Read-write. Cold reset: 1. 0=Hwm function defined at <a href="#">3.25.14 [Power Management Block 2 (PM2) Registers]</a> is reset by RsmRst. 1=Hwm function defined at <a href="#">3.25.14 [Power Management Block 2 (PM2) Registers]</a> is reset by PciRst.
5	<b>SlpTypeControl.</b> Read-write. Cold reset: 1. 1=Enable the function of <a href="#">AcpiPm1CntBlkx00[SlpTypeEn]</a> . 0= <a href="#">AcpiPm1CntBlkx00[SlpTypeEn]</a> bit has no effect.
4	<b>KbRstEn.</b> Read-write. Cold reset: 1. BIOS: See <a href="#">Table 238</a> . 1= Enable KBRST# pin to trigger keyboard reset.

3:2	<b>CpuRstControl.</b> Read-write. Cold reset: 0. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>CpuReset is deasserted after PciReset</td> </tr> <tr> <td>01b</td> <td>CpuReset is deasserted as PciReset</td> </tr> <tr> <td>10b</td> <td>CpuReset is deasserted before PciReset</td> </tr> <tr> <td>11b</td> <td>CpuReset is deasserted after PciReset</td> </tr> </tbody> </table>	Bits	Definition	00b	CpuReset is deasserted after PciReset	01b	CpuReset is deasserted as PciReset	10b	CpuReset is deasserted before PciReset	11b	CpuReset is deasserted after PciReset
Bits	Definition										
00b	CpuReset is deasserted after PciReset										
01b	CpuReset is deasserted as PciReset										
10b	CpuReset is deasserted before PciReset										
11b	CpuReset is deasserted after PciReset										
1	<b>KbPciRstEn.</b> Read-write. Cold reset: 1. This bit must not be programmed by the BIOS. It should be left with the power up default value of 1. 1=Make PCI reset during keyboard reset, which can be triggered by KBRST# pin or IMC.										
0	<b>SoftResetEn.</b> Read-write. Cold reset: 0. 1=Block any reset request until the system is not in C state.										

### PMxC0 S5/Reset Status

This register shows the source of previous reset.

Bits	Description										
31:30	Reserved.										
29	<b>EcWatchdogRst.</b> Write-1-to-clear; Read. Cold reset: 0.										
28	<b>HangReset.</b> Write-1-to-clear; Read. Cold reset: 0.										
27	<b>SyncFlood.</b> Write-1-to-clear; Read. Cold reset: 0.										
26	<b>RemoteResetFromASF.</b> Write-1-to-clear; Read. Cold reset: 0.										
25	<b>WatchdogIssueReset.</b> Write-1-to-clear; Read. Cold reset: 0.										
24	<b>FailBootRst.</b> Write-1-to-clear; Read. Cold reset: 0.										
23	<b>LtReset.</b> Write-1-to-clear; Read. Cold reset: 0.										
22	<b>KbReset.</b> Write-1-to-clear; Read. Cold reset: 0.										
21	<b>SleepReset.</b> Write-1-to-clear; Read. Cold reset: 0.										
20	<b>DoFullReset.</b> Write-1-to-clear; Read. Cold reset: 0.										
19	<b>DoReset.</b> Write-1-to-clear; Read. Cold reset: 0.										
18	<b>DoInit.</b> Write-1-to-clear; Read. Cold reset: 0.										
17	<b>SoftPciRst.</b> Write-1-to-clear; Read. Cold reset: 0.										
16	<b>UserRst.</b> Write-1-to-clear; Read. Cold reset: 0.										
15:14	<b>PmeTurnOffTime.</b> Read-write. Cold reset: 0. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>1 ms</td> </tr> <tr> <td>01b</td> <td>2 ms</td> </tr> <tr> <td>10b</td> <td>4 ms</td> </tr> <tr> <td>11b</td> <td>8 ms</td> </tr> </tbody> </table>	Bits	Definition	00b	1 ms	01b	2 ms	10b	4 ms	11b	8 ms
Bits	Definition										
00b	1 ms										
01b	2 ms										
10b	4 ms										
11b	8 ms										
13	<b>DisSbToNbPG.</b> Read-write. Cold reset: 0. 1=Disable NBPwrGood.										
12	<b>DisableLdtPwrGood.</b> Read-write. Cold reset: 0. This is a LdtPwrGood control bit. 1=Disable the LdtPwrGood assertion along with NBPwrGood.										
11	<b>SlpS3ToLdtPwrGdEn.</b> Read-write. Cold reset: 1. 1=De-assert LDT_PWRGD as long as SLP_S3# goes low.										

10	<b>PwrGdDwnBeforeSlpS3</b> . Read-write. Cold reset: 0. BIOS: 1. 1=Delay SLP_S3 by 64 us and also qualify the FCH PwrGood with SLP_S3; This allows internal logic to put signals into correct states before turning off the S0 power.
9:6	Reserved.
5	<b>ShutDownFan0</b> . Write-1-to-clear; Read. Cold reset: 0.
4	<b>RemotePowerDownFromASF</b> . Write-1-to-clear; Read. Cold reset: 0.
3	<b>ThermalTripFromTemp</b> . Write-1-to-clear; Read. Cold reset: 0.
2	<b>Shutdown</b> . Write-1-to-clear; Read. Cold reset: 0.
1	<b>FourSecondPwrBtn</b> . Write-1-to-clear; Read. Cold reset: 0.
0	<b>ThermalTrip</b> . Write-1-to-clear; Read. Cold reset: 0.

### PMxC4 ResetCommand

Bits	Description
7	<b>ResetEn</b> . Read-write. Cold reset: 0. 0=Writing to bit Reset is not allowed. 1=Writing to bit Reset is allowed.
6	<b>ResetAllAcpi</b> . Write-1-only; cleared-by-hardware. Cold reset: 0. Writing 1 to this bit emulates a reset button event.
5	<b>ResetButtonEn</b> . Read-write. Cold reset: 1. 1=Enable ResetButton. Sys_Reset_L assertion will trigger reset. 0=Disable ResetButton. Set to 0b if the System_rst_L pin is not used or configured to Gevent19/gpio73 (function 1) in IOMUX 73h.
4	<b>ResetPcie</b> . Read-write. Cold reset: 0. 1=Reset the GPP ports. 0=Release the PCIe reset.
3	<b>UsrRst2Pll</b> . Read-write. Cold reset: 1. 1=Stop PLL when reset button is pressed.
2	<b>SelectDebug</b> . Read-write. Cold reset: 0. 0=Select <a href="#">PMxC0 [S5/Reset Status]</a> to be S5/Reset Status register. 1=Select <a href="#">PMxC0 [S5/Reset Status]</a> to be a debug status register.
1	<b>MemRstDisable</b> . Read-write. Cold reset: 0. 1=The memory reset function at DDR_RST# pin is disabled.
0	<b>Reset</b> . Write-1-only; cleared-by-hardware. Cold reset: 0. Writing 1 to this bit causes a PCI reset. This bit is enabled by RestEn bit.

### PMxC5 CF9 Shadow

Bits	Description
7:0	Alias of <a href="#">IOCF9</a> .

### PMxC8 Misc

Bits	Description
31:24	<b>ClkIntrVectorOrd</b> . Read-write. Cold reset: 0. Specifies the value used to identify the clock interrupt.
23	Reserved.
22	<b>ClkIntrVectorOrdEn</b> . Read-write. Cold reset: 0. 1=The system timer interrupt in the IOAPIC is tagged with a value defined by ClkIntrVectorOrd.

21	Reserved.										
20	<b>ProcHotStsEn.</b> Read-write. Cold reset: 0. 1=Enable PROCHOT# to generate TwarnStatus SMIx84[16] and thermal throttle.										
19	<b>UseCpuRst.</b> Read-write. Cold reset: 1. 0=System reset causes INIT# instead of CPURST#.										
18	<b>UseBypassRom.</b> Read-write. Cold reset: 0. 1=Override the ROM straps and use BypassRomSel to determine which type of ROM to use. This is for BIOS debugging purpose or for systems having multiple BIOSes on board.										
17:16	<b>BypassRomSel.</b> Read-write. Cold reset: 0. These two bits override the two ROM strap pins if [UseBypassRom] == 1. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>LPC ROM</td> </tr> <tr> <td>01b</td> <td>Reserved</td> </tr> <tr> <td>10b</td> <td>Reserved</td> </tr> <tr> <td>11b</td> <td>SPI ROM</td> </tr> </tbody> </table>	Bits	Definition	00b	LPC ROM	01b	Reserved	10b	Reserved	11b	SPI ROM
Bits	Definition										
00b	LPC ROM										
01b	Reserved										
10b	Reserved										
11b	SPI ROM										
15	<b>HideSmbus.</b> Read-write. Cold reset: 0. 1=Hide the SMBus PCI configuration space and promote LPC bridge PCI configuration space to function 0.										
14	Reserved.										
13	<b>IdChangeEn.</b> Read-write. Cold reset: 0. 1=Allow the software to change D14F0x00[DeviceID] and D14F0x08[RevisionID].										
12	<b>S5ResetOverride.</b> Read-write. Cold reset: 0. 1=Mask off internet PCI reset used in ACPI.										
11	<b>WriteBackEnable.</b> Read-write. Cold reset: 0. 1=The WakeOnRing status bit is written back to HD audio controller upon system power up.										
10	<b>LlbEn.</b> Read-write. Cold reset: 0. 1=LLB function is enabled and system won't wake up from ACPI S state until LLB# is de-asserted.										
9:8	<b>TempPolarity.</b> Read-write. Cold reset: 0. Temperature polarity control for THRMTRIP and TALERT respectively. 0=Active low. 1=Active high.										
7	<b>DisablePciRom.</b> Read-write. Cold reset: 0. 1=Disable PCI from strap.										
6	<b>TwarnEn.</b> Read-write. Cold reset: 0. 1=Enable TALERT# pin.										
5	<b>EPromDeSelect.</b> Read-write. Cold reset: 0.										
4	<b>UseAcpiStraps.</b> Read-write. Cold reset: 0.										
3	<b>TDeadEn.</b> Read-write. Cold reset: 1. 1=GEVENT2 takes up the THRMTRIP function; When THRMTRIP pin is low and [TFatalEn] is set, hardware switches the system to S5 automatically.										
2	<b>TFatalEn.</b> Read-write. Cold reset: 1. 1=Enable both the soft PciRst and the THRMTRIP function.										
1	Reserved.										
0	<b>CpuIoPullDownDrvStrength.</b> Read-write. Cold reset: 0. 1=The integrated pull-down drive strength of all CPU IOs are increased by 50%.										

**PMxCC IoDrvSth**

Bits	Description
31:30	Reserved.
29:27	<b>IoDrvSth_Misc.</b> Read-write. Cold reset: 011b. IO drive strength for GA20, KBRST#, SERIRQ, and SATA_ACT# pads.

26:24	<b>IoDrvSth_Gpio.</b> Read-write. Cold reset: 011b. IO drive strength for GPIO3, and GPIO[48:52] pads.
23:21	<b>IoDrvSth_Gpio.</b> Read-write. Cold reset: 011b. IO drive strength for BMREQ#, GPIO[0, 2, 4, 5, 7, 8, 9, 13, 37, 38, 39, 40], GPOC[0, 1] pads.
20:18	<b>IoDrvSth_Req.</b> Read-write. Cold reset:011b. IO drive strength for REQ[3:0]# when they are configured as GPIO.
17:15	Reserved.
14:12	<b>IoDrvSth_LPC.</b> Read-write. Cold reset: 011b. IO drive strength for LPC LAD, LFRAME# pads. The recommended setting for single load is 111b.
11:0	Reserved.

### PMxD0 RstCntrl

Bits	Description
7:6	<b>RstLengthSel.</b> Read-write. Cold reset: 0. This field selects which register is defined at <a href="#">PMxD1</a> .
5:0	Reserved.

### PMxD1 Reset Function

Address: [PMxD0](#)[RstLengthSel].

Bits	Description
7:0	<b>RstFunctLength.</b>

### PMxD1\_x0 RstLength

Bits	Description
7:0	<b>RstLength.</b> Read-write. Cold reset: F4h. Specifies the A_RST# length. The amount of reset time = $RstLength * 4096 * 69.84 \text{ ns}$ .

### PMxD1\_x1 APURstLength

Bits	Description
7:0	<b>APURstLength.</b> Read-write. Cold reset: FBh. Specifies the APU_RST# length. The amount of reset time = $ApuRstLength * 4096 * 69.84 \text{ ns}$ .

### PMxD1\_x3 APUPwrGdLength

Bits	Description
7:0	<b>APUPwrGdLength.</b> Read-write. Cold reset: EAh. Specifies the LdtPwrGd latency. The amount of delay = $APUPwrGdLength * 4096 * 69.84 \text{ ns}$ .



**PMxD2 PmioDebug**

Bits	Description
7:6	Reserved.
5:4	<b>LpcClkDrvSth</b> . Read-write. Cold reset: 00b. Drive strength control for LpcClk[1:0] respectively. 0=Clock output is 4 mA. 1=Clock output is 8 mA.
3:0	Reserved.

**PMxD3 ManualReset**

Bits	Description
7	Reserved.
6	<b>AssertSDRstB</b> . Read-write. Cold reset: 1. 0=Assert the reset to SDIO controller.
5	<b>AssertAzRstB</b> . Read-write. Cold reset: 1. 0=Assert the reset to AZ controller.
4	<b>AssertUsbRstB</b> . Read-write. Cold reset: 1. BIOS: See 2.15.2.5. 0=Assert the reset to USB controllers.
3	<b>AssertSataRstB</b> . Read-write. Cold reset: 1. 0=Assert the reset to SATA controller.
2	<b>AssertPciRstB</b> . Read-write. Cold reset: 1. 0=Assert PCIRST# low.
1	<b>AssertARstB</b> . Read-write. Cold reset: 1. 0=Assert A_RST# low.
0	<b>AssertLdtRstB</b> . Read-write. Cold reset: 1. 0=Assert LDT_RST# low.

**PMxD6 IMC Gating**

Bits	Description
15	<b>ImcEnable</b> . Read-write. Reset: 0. BIOS: See 2.15.8. 1=Enable IMC.
14:0	Reserved.

**PMxD8 Eprom Index**

Bits	Description
7:0	<b>EpromIndex</b> . Read-write. Cold reset: 0.

**PMxD9 Eprom Data**

Bits	Description
7:0	<b>EpromData</b> . Read-write. Cold reset: 0.

**PMxDA SataConfig**

Bits	Description										
15:8	Reserved.										
7:6	<b>RefDivSel.</b> Read-write. Cold reset: 00b. BIOS: See 2.15.3.5. This specifies the reference clock divider setting. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Divide by 1 (25 MHz or 48 MHz reference clock)</td> </tr> <tr> <td>01b</td> <td>Divide by 2</td> </tr> <tr> <td>10b</td> <td>Divide by 4 (100 MHz reference clock)</td> </tr> <tr> <td>11b</td> <td>Divide by 4 (100 MHz reference clock)</td> </tr> </tbody> </table>	Bits	Definition	00b	Divide by 1 (25 MHz or 48 MHz reference clock)	01b	Divide by 2	10b	Divide by 4 (100 MHz reference clock)	11b	Divide by 4 (100 MHz reference clock)
Bits	Definition										
00b	Divide by 1 (25 MHz or 48 MHz reference clock)										
01b	Divide by 2										
10b	Divide by 4 (100 MHz reference clock)										
11b	Divide by 4 (100 MHz reference clock)										
5:4	<b>RefClkSel.</b> Read-write. Cold reset: 01b. BIOS: See 2.15.3.5. This specifies the reference clock source selection for SATA PLL. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Reference clock from crystal oscillator via PAD_XTALI and PAD_XTALO</td> </tr> <tr> <td>01b</td> <td>Reference clock from internal clock through CP_PLL_REFCLK_P and CP_PLL_REFCLK_N via RDL</td> </tr> <tr> <td>10b</td> <td>Reserved.</td> </tr> <tr> <td>11b</td> <td>Reserved.</td> </tr> </tbody> </table>	Bits	Definition	00b	Reference clock from crystal oscillator via PAD_XTALI and PAD_XTALO	01b	Reference clock from internal clock through CP_PLL_REFCLK_P and CP_PLL_REFCLK_N via RDL	10b	Reserved.	11b	Reserved.
Bits	Definition										
00b	Reference clock from crystal oscillator via PAD_XTALI and PAD_XTALO										
01b	Reference clock from internal clock through CP_PLL_REFCLK_P and CP_PLL_REFCLK_N via RDL										
10b	Reserved.										
11b	Reserved.										
3	Reserved.										
2	<b>SetMaxGen2.</b> Read-write. Cold reset: 0. 0=SATA controller operates in maximum Gen3 (6.0 Gb/s) speed. 1=SATA controller operates in maximum Gen2 (3.0 Gb/s) speed for reduced power consumption.										
1	Reserved.										
0	<b>SataEnable.</b> Read-write. Cold reset: 1. BIOS: 1. 0=SATA controller is disabled. 1=SATA controller is enabled. Whenever SATA controller is being enabled by programming this field from 0 to 1, a <b>IOCF9</b> software reset is recommended to reset the controller to a proper operational state.										

**PMxDC SataConfig2**

Bits	Description
31:17	Reserved.
16	<b>PllCalibEn: SATA PHY PLL calibration enable.</b> Read-write. Reset: 0. BIOS: See 2.15.3.3.
15:0	Reserved.

**PMxE0 ABRegBar**

Bits	Description
31:16	Reserved.
15:3	<b>ABRegBar.</b> Read-write. Cold reset: 0. BIOS: See 2.15.10. IO Base address of AB Configuration Registers. See AB configuration registers defined in 3.25.2 [AB Configuration Registers].
2:0	Reserved.

**PMxE4 ABDebug**

Bits	Description										
31:2											
1:0	<b>BlinkControl.</b> Read-write. Cold reset: 0. This register controls the BLINK pin behavior. BLINK pin can be used to control the on/off of a LED on the board. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Disabled. BLINK pin can be controlled by GPIO logic if so configured.</td> </tr> <tr> <td>01b</td> <td>BLINK pin is repeatedly driven to low for 1 sec and then released for 3 sec.</td> </tr> <tr> <td>10b</td> <td>BLINK pin is repeatedly driven to low for 2 sec and then released for 2 sec.</td> </tr> <tr> <td>11b</td> <td>BLINK pin is always driven to low.</td> </tr> </tbody> </table>	Bits	Definition	00b	Disabled. BLINK pin can be controlled by GPIO logic if so configured.	01b	BLINK pin is repeatedly driven to low for 1 sec and then released for 3 sec.	10b	BLINK pin is repeatedly driven to low for 2 sec and then released for 2 sec.	11b	BLINK pin is always driven to low.
Bits	Definition										
00b	Disabled. BLINK pin can be controlled by GPIO logic if so configured.										
01b	BLINK pin is repeatedly driven to low for 1 sec and then released for 3 sec.										
10b	BLINK pin is repeatedly driven to low for 2 sec and then released for 2 sec.										
11b	BLINK pin is always driven to low.										

**PMxE8 SDFlashCntrl**

Bits	Description
7:2	Reserved.
1	Reserved.
0	<b>SDFlashEnable.</b> Read-write. Cold reset: 0. 0=Disable SD flash controller. 1=Enable SD flash controller, and GPIO[73:80] become SD flash interface.

**PMxEB AzEn**

Bits	Description
7:2	Reserved.
1	<b>AzNoSnoopEnable.</b> Read-write. Cold reset: 0. 1=HD audio data transfer doesn't cause the <a href="#">AcpiPmEvtBlkx00[BmStatus]</a> bit to be set nor to wake up the CPU from C3 state. Under current C1e implementation, there is no need to set this bit.
0	<b>AzEnable.</b> Read-write. Cold reset: 1. BIOS: 1. 0=Disable HD audio controller. 1=Enable HD audio controller.

**PMxEC LpcGating**

Bits	Description
7:3	Reserved.
2	<b>AbNoBypassEn.</b> Read-write. Cold reset: 0. BIOS: 1. 1=Tells the A-Link that the LPC cycle should not be bypassed when a retry has timed out.
1	<b>LpcA20En.</b> Read-write. Cold reset: 0. 1=Enable A20# input.
0	<b>LpcEnable.</b> Read-write. Cold reset: 1. 1=Enable LPC bridge.

**PMxED USB Gating**

Bits	Description
7:5	Reserved.
4	<b>UsbSmiEn.</b> Read-write. Cold reset: 0. 1=Enable SMI for USB legacy support.
3:2	Reserved.
1	<b>UsbIrqEn.</b> Read-write. Cold reset: 0. 1=Route IRQ1/IRQ12 from USB OHCI to PIC/IOAPIC for USB legacy support.
0	<b>UsbA20En.</b> Read-write. Cold reset: 0. 1=Enable A20Gate from USB OHCI controller for USB legacy support.

**PMxEF USB Enable**

BIOS: See 2.15.2.4 [Enabling the xHCI Controller] && 2.15.2.15 [xHCI Enable OHCI3/EHCI3 on S4/S5 State Entry].

Bits	Description
7	<b>PortRoutingSelect.</b> Read-write. Cold reset: 0. Specifies controller(s) for USB ports 8 and 9. 0=EHCI 3 and OHCI 3 controllers. 1=xHCI controller.
6	Reserved.
5	<b>Usb3EhciEnable.</b> Read-write. Cold reset: 1. 1=Enable EHCI 3 controller.
4	<b>Usb3OhciEnable.</b> Read-write. Cold reset: 1. 1=Enable OHCI 3 controller (device 16h, function 0).
3	<b>Usb2EhciEnable.</b> Read-write. Cold reset: 1. 1=Enable EHCI 2 controller.
2	<b>Usb2OhciEnable.</b> Read-write. Cold reset: 1. 1=Enable OHCI 2 controller (device 13h, function 0).
1	<b>Usb1EhciEnable.</b> Read-write. Cold reset: 1. 1=Enable EHCI 1 controller.
0	<b>Usb1OhciEnable.</b> Read-write. Cold reset: 1. 1=Enable OHCI 1 controller (device 12h, function 0).

**PMxF0 USB Control****Table 239: BIOS Recommendations for UsbPhyS5PwrDwnEnable**

Option	Description	Setting	Comment
1	USB Wake from S5 not supported on the platform	1	When the USB power rails USB PHY PLL, USB PHY core power and USB PHY DLL are connected to S0-S3 power, set the bit to 1 to disable the USB S4/S5 wake up function
2	USB Wake from S5 supported on the platform	0	When the USB power rails USB PHY PLL, USB PHY core power and USB PHY DLL are connected to S5 power, set the bit to 0 to enable the USB S4/S5 wakeup function

Bits	Description
15	<b>PmioEhciMemSlpDis.</b> Read-write. Cold Reset: 0. 0=Enable EHCI memory sleep. 1=Disable EHCI memory sleep.

14:13	Reserved.																		
12	<b>Usb2BtGlobalClkGateEn.</b> Read-write. Cold reset: 1. 1=Enable USB2.0 B-Link Global Clock Gating.																		
11	Reserved.																		
10:8	<p><b>UsbSleepCtrl.</b> Read-write. Cold reset: 3h. Control on USB advanced asynchronous sleep function. Setting of 000b:100b are for the advanced asynchronous sleep.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>Standard 10 us sleep</td> </tr> <tr> <td>001b</td> <td>Advanced sleep for up to 2 micro frames</td> </tr> <tr> <td>010b</td> <td>Advanced sleep for up to 4 micro frames</td> </tr> <tr> <td>011b</td> <td>Advanced sleep for up to 6 micro frames</td> </tr> <tr> <td>100b</td> <td>Advanced sleep for up to next micro frame</td> </tr> <tr> <td>101b</td> <td>Reserved</td> </tr> <tr> <td>110b</td> <td>Reserved</td> </tr> <tr> <td>111b</td> <td>If CPU is in C state and the controller has already exhausted the link list, it can simply stop the asynchronous packets until CPU resumes back to C0 state. In this case, the controller will resume back to its standard mode.</td> </tr> </tbody> </table>	Bits	Definition	000b	Standard 10 us sleep	001b	Advanced sleep for up to 2 micro frames	010b	Advanced sleep for up to 4 micro frames	011b	Advanced sleep for up to 6 micro frames	100b	Advanced sleep for up to next micro frame	101b	Reserved	110b	Reserved	111b	If CPU is in C state and the controller has already exhausted the link list, it can simply stop the asynchronous packets until CPU resumes back to C0 state. In this case, the controller will resume back to its standard mode.
Bits	Definition																		
000b	Standard 10 us sleep																		
001b	Advanced sleep for up to 2 micro frames																		
010b	Advanced sleep for up to 4 micro frames																		
011b	Advanced sleep for up to 6 micro frames																		
100b	Advanced sleep for up to next micro frame																		
101b	Reserved																		
110b	Reserved																		
111b	If CPU is in C state and the controller has already exhausted the link list, it can simply stop the asynchronous packets until CPU resumes back to C0 state. In this case, the controller will resume back to its standard mode.																		
7:6	Reserved.																		
5	<b>PmioOhciMemSlpDis.</b> Read-write. Cold reset: 0. 0=Enable OHCI memory sleep mode. 1=Disable OHCI memory sleep mode.																		
4	<b>Usb11PdResistorEnable.</b> Read-write. Cold reset: 1. 0=Disconnect pull-down resistors on stand-alone USB1.1 pads.																		
3	<b>UsbS5ResetEnable.</b> Read-write. Cold reset: 1. 1=Enable USB reset on S4/S5 resume detection.																		
2	<b>UsbKbResetEnable.</b> Read-write. Cold reset: 1. BIOS: See 2.15.2.3. 1=Enable resetting USB on keyboard reset.																		
1	Reserved.																		
0	<b>UsbPhyS5PwrDwnEnable.</b> Read-write. Cold reset: 0. BIOS: See Table 239. 1=Enable S4/S5 USB PHY power down support and disable S4 USB wake up support. 0=Disable S4/S5 USB PHY power down support and enable S4 USB wakeup support. The bit has to be set to 0 to support S4 USB wake up.																		

### 3.25.14 Power Management Block 2 (PM2) Registers

PM2 register space is accessed through two methods:

- Indirect IO access through index/data address pair at [IOCD0 \[PM2\\_Index\]](#) and [IOCD1 \[PM2\\_Data\]](#). Software first programs the offset into the index register [IOCD0 \[PM2\\_Index\]](#) and then reads/writes to/from the data register [IOCD1 \[PM2\\_Data\]](#).
- Direct memory mapped or IO mapped access through the AcpiMmio region. The PM2 registers range from [PMx24\[AcpiMmioAddr\]+400h](#) to [PMx24\[AcpiMmioAddr\]+4FFh](#). See [PMx24 \[AcpiMmioEn\]](#).

#### PM2x00 Fan0InputControl

Bits	Description
7:4	Reserved.

3	<b>TwoRampAlgorithmEn.</b> Read-write. Reset: 0. 1=The two ramp fan control algorithm is enabled. 0=Disabled.										
2:0	<b>FanInputControl.</b> Read-write. Reset: 0. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>100b-000b</td> <td>Reserved.</td> </tr> <tr> <td>101b</td> <td>FanOut0 is enabled and temperature input is from TempTsi.</td> </tr> <tr> <td>110b</td> <td>FanOut0 is enabled and temperature input is 0.</td> </tr> <tr> <td>111b</td> <td>FanOut0 is disabled.</td> </tr> </tbody> </table>	Bits	Definition	100b-000b	Reserved.	101b	FanOut0 is enabled and temperature input is from TempTsi.	110b	FanOut0 is enabled and temperature input is 0.	111b	FanOut0 is disabled.
Bits	Definition										
100b-000b	Reserved.										
101b	FanOut0 is enabled and temperature input is from TempTsi.										
110b	FanOut0 is enabled and temperature input is 0.										
111b	FanOut0 is disabled.										

### PM2x01 Fan0Control

When [AutoMode] == 1, the active duty cycle is controlled by the hardware automatically.

- If ActualTemperature < LowTemp{PM2x07, PM2x06}, DutyCycle = 0;
- If LowTemp{PM2x07, PM2x06} <= ActualTemperature < MedTemp{PM2x09, PM2x08}, DutyCycle = PM2x03 [LowDuty0].
- If MedTemp{PM2x09, PM2x08} <= ActualTemperature < HighTemp{PM2x0B, PM2x0A},
  - If [LinearMode] == 0, DutyCycle = PM2x04 [MedDuty0].
  - If [LinearMode] == 1, DutyCycle = ((ActualTemperature - LowTemp{PM2x07, PM2x06}) \* (PM2x05[Multiplier] + 1) >> PM2x05[DutySel]) + PM2x03 [LowDuty0].
- If ActualTemperature >= HighTemp{PM2x0B, PM2x0A}, DutyCycle = 100%.

In Automode, hysteresis limit defined by PM2x0C [LinearRange0] is applied to keep the fan from oscillating erratically.

Bits	Description
7:3	<b>LinearAdjust.</b> Read-write. Reset: 0. Specifies the additional offset to effective duty cycle under Linear mode.
2	<b>FanPolarity.</b> Read-write. Reset: 0. 0=FanOut0 drives low. 1=FanOut0 drives high.
1	<b>LinearMode.</b> Read-write. Reset: 0. 0=Use step function. 1=Use Linear function. See above description for details.
0	<b>AutoMode.</b> Read-write. Reset: 0. 0=FanOut0 is controlled by PM2x03 [LowDuty0]. 1=FanOut0 is controlled by the temperature input. See the above description for details on AutoMode.

### PM2x02 Fan0Freq

Bits	Description																																				
7:0	<b>FanFreq.</b> Read-write. Reset: 0. Normally, 4-wire fan runs at 25 kHz and 3-wire fan runs at 100 Hz. FanOut0 frequency is programmed as below: <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00h</td> <td>28.64 kHz</td> <td>F9h</td> <td>58 Hz</td> </tr> <tr> <td>01h</td> <td>25.78 kHz</td> <td>FAh</td> <td>44 Hz</td> </tr> <tr> <td>02h</td> <td>23.44 kHz</td> <td>FBh</td> <td>35 Hz</td> </tr> <tr> <td>03h</td> <td>21.48 kHz</td> <td>FCh</td> <td>29 Hz</td> </tr> <tr> <td>04h</td> <td>19.83 kHz</td> <td>FDh</td> <td>22 Hz</td> </tr> <tr> <td>05h</td> <td>18.41 kHz</td> <td>FEh</td> <td>14 Hz</td> </tr> <tr> <td>F7h</td> <td>100 Hz</td> <td>FFh</td> <td>11 Hz</td> </tr> <tr> <td>F8h</td> <td>87 Hz</td> <td>F6h-06h</td> <td>1/(FanFreq*2048*15 ns)</td> </tr> </tbody> </table>	Bits	Definition	Bits	Definition	00h	28.64 kHz	F9h	58 Hz	01h	25.78 kHz	FAh	44 Hz	02h	23.44 kHz	FBh	35 Hz	03h	21.48 kHz	FCh	29 Hz	04h	19.83 kHz	FDh	22 Hz	05h	18.41 kHz	FEh	14 Hz	F7h	100 Hz	FFh	11 Hz	F8h	87 Hz	F6h-06h	1/(FanFreq*2048*15 ns)
Bits	Definition	Bits	Definition																																		
00h	28.64 kHz	F9h	58 Hz																																		
01h	25.78 kHz	FAh	44 Hz																																		
02h	23.44 kHz	FBh	35 Hz																																		
03h	21.48 kHz	FCh	29 Hz																																		
04h	19.83 kHz	FDh	22 Hz																																		
05h	18.41 kHz	FEh	14 Hz																																		
F7h	100 Hz	FFh	11 Hz																																		
F8h	87 Hz	F6h-06h	1/(FanFreq*2048*15 ns)																																		

**PM2x03 LowDuty0**

Bits	Description								
7:0	<p><b>LowDuty0.</b> Read-write. Reset: 0. Specifies Fan0 duty number if (LowTemp{PM2x07, PM2x06} &lt;= temperature &lt; MedTemp{PM2x09, PM2x08}). There are 256 time slots in one fan cycle. Duty number N represents the (N+1)th time slot. Fan actively spins in time slot 0~ slot N, and stops from slot (N+1) ~ slot 255.</p> <table border="0"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>00h</td> <td>Always stop</td> </tr> <tr> <td>FEh-01h</td> <td>Fan spins for time slots &lt;= LowDuty, stops for time slots &gt; LowDuty.</td> </tr> <tr> <td>FFh</td> <td>Full speed run</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	00h	Always stop	FEh-01h	Fan spins for time slots <= LowDuty, stops for time slots > LowDuty.	FFh	Full speed run
<u>Bits</u>	<u>Definition</u>								
00h	Always stop								
FEh-01h	Fan spins for time slots <= LowDuty, stops for time slots > LowDuty.								
FFh	Full speed run								

**PM2x04 MedDuty0**

Bits	Description								
7:0	<p><b>MedDuty.</b> Read-write. Reset: 0. Specifies Fan0 duty number using step function if (Med-Temp{PM2x09, PM2x08} &lt;= temperature &lt;= HighTemp{PM2x0B, PM2x0A}). There are 256 time slots in one fan cycle. Duty number N represents the (N+1)th time slot. Fan actively spins in time slot0 ~ slotN, and stops from slot (N+1) ~ slot255.</p> <table border="0"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>00h</td> <td>Always stop</td> </tr> <tr> <td>FEh-01h</td> <td>Fan spins for time slots &lt;= MedDuty, stops for time slots &gt; MedDuty.</td> </tr> <tr> <td>FFh</td> <td>Full speed run</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	00h	Always stop	FEh-01h	Fan spins for time slots <= MedDuty, stops for time slots > MedDuty.	FFh	Full speed run
<u>Bits</u>	<u>Definition</u>								
00h	Always stop								
FEh-01h	Fan spins for time slots <= MedDuty, stops for time slots > MedDuty.								
FFh	Full speed run								

**PM2x05 Multiplier0**

When (PM2x01[AutoMode] == 1) && (PM2x00[TwoRampAlgorithm] == 1), Fan0 is in TwoRamp mode. The slope value of ramp1 and ramp0 is: Slope[7:0] = {PM2x05[DutySel], PM2x05[Multiplier]}. Slope[7:2] are the integer bits and Slope[1:0] are the fractional bits of the ramp slope.

BIOS has to calculate Slope[7:0] using the following equation:

$$\text{Slope} = (16'hFF00 - \{\text{LowDuty0}[7:0], 8'b0\}) / (\text{HighTemp0}[15:0] - \text{Hysteresis0}[15:0] - \text{MedTemp0}[15:0])$$

For example, if our setting is:

$$\text{LowDuty0}[7:0] = 25\% = 40h$$

$$\text{HighTemp0}[15:0] = 90^{\circ}\text{C} = 5A00h$$

$$\text{MedTemp0}[15:0] = 40^{\circ}\text{C} = 2800h$$

$$\text{Hysteresis0}[15:0] = 10^{\circ}\text{C} = 0A00h$$

Then:

$$\text{Slope} = (\text{FF00h} - 4000h) / (5A00h - 0A00h - 2800h) = 48896 / 10240 = 4.775$$

Convert the number 4.775 into our 8-bit format: 00010011b (13h). BIOS should program the Multiplier0 register as 13h in this example.

Bits	Description
------	-------------

7:6	<b>DutySel.</b> Read-write. Reset: 0. When Fan0 is programmed in AutoMode with linear function being selected, this field selects part of duty to be fed into fan as described in <a href="#">PM2x01 [Fan0Control]</a> register description. When Fan0 is programmed in TwoRamp mode, this field specifies the upper 2 bits in Slope[7:0]. See the above register description for detail.
5:0	<b>Multiplier.</b> Read-write. Reset: 0. When Fan0 is programmed in AutoMode with linear function being selected, this field specifies the factor to calculate duty number as described in <a href="#">PM2x01 [Fan0Control]</a> register description. When Fan0 is programmed in TwoRamp mode, this field specifies the lower 6 bits in Slope[7:0]. See the above register description for detail.

**PM2x06 LowTemp0Lo**

Bits	Description
7:0	<b>LowTemp0[7:0].</b> Read-write. Reset: 0. Specifies the lower 8 bits of low temperature threshold.

**PM2x07 LowTemp0Hi**

Bits	Description
7:0	<b>LowTemp0[15:8].</b> Read-write. Reset: 0. Specifies the higher 8 bits of low temperature threshold.

**PM2x08 MedTemp0Lo**

Bits	Description
7:0	<b>MedTemp0[7:0].</b> Read-write. Reset: 0. Specifies the lower 8 bits of medium temperature threshold.

**PM2x09 MedTemp0Hi**

Bits	Description
7:0	<b>MedTemp0[15:8].</b> Read-write. Reset: 0. Specifies the higher 8 bits of medium temperature threshold.

**PM2x0A HighTemp0Lo**

Bits	Description
7:0	<b>HighTemp0[7:0].</b> Read-write. Reset: 0. Specifies the lower 8 bits of high temperature threshold.

**PM2x0B HighTemp0Hi**

Bits	Description
7:0	<b>HighTemp0[15:8].</b> Read-write. Reset: 0. Specifies the higher 8 bits of high temperature threshold.



**PM2x0C LinearRange0**

Bits	Description
7:0	<b>LinearRange</b> . Read-write. Reset: 0. Specifies a variable range that Fan0 can tolerate. Fan0 is not affected if temperature varies within this range.

**PM2x0D LinearHoldCount0**

Bits	Description
7:0	<b>LinearHoldCount</b> . Read-write. Reset: 0. Specifies the fan cycles to be waited before duty cycle can be changed.

**PM2x0E Fan0Hysteresis**

Bits	Description
15:8	<b>HysteresisHi</b> . Read-write. Reset: 0. Specifies the hysteresis value (in temperature) of the Two Ramp Fan Control Algorithm. The unit is °C. See <a href="#">PM2x05 [Multiplier0]</a> on how to program this register.
7:0	<b>HysteresisLo</b> . Read-write. Reset: 0. Reserved. This byte should always be programmed as 0.

**PM2x50 Med2Temp0Lo**

Bits	Description
7:0	<b>Med2Temp0[7:0]</b> . Read-write. Reset: 0. IF ( <a href="#">PM2x00[TwoRampAlgorithmEn]</a> ==1) THEN It specifies the lower byte of the temperature value of the turning point on the ramp. ELSE Unused. ENDIF.

**PM2x51 Med2Temp0Hi**

Bits	Description
7:0	<b>Med2Temp0[15:8]</b> . Read-write. Reset: 0. IF ( <a href="#">PM2x00[TwoRampAlgorithmEn]</a> ==1) THEN It specifies the higher byte of the temperature value of the turning point on the ramp. ELSE Unused. ENDIF.

**PM2x52 Med2Duty0**

Bits	Description
7:0	<b>Med2Duty0[7:0]</b> . Read-write. Reset: 0. IF ( <a href="#">PM2x00[TwoRampAlgorithmEn]</a> ==1) THEN It specifies the fan duty value of the turning point on the ramp. ELSE Unused. ENDIF.

**PM2x53 Multiplier2\_0**

BIOS has to calculate [Multiplier2\\_0\[7:0\]](#) using the following equation:

$$\text{Multiplier2\_0} = (\{\text{Med2Duty0}[7:0], 8'b0\} - \{\text{LowDuty0}[7:0], 8'b0\}) / (\text{Med2Temp0}[15:0] - \text{MedTemp0}[15:0])$$

For example, if our setting is:

[PM2x52\[Med2Duty0\[7:0\]\]](#)=50%=80h

PM2x03[LowDuty0[7:0]]=25%=40h

Med2Temp0[15:0]={PM2x51, PM2x50}=70°C=4600h

MedTemp0[15:0]={PM2x09, PM2x08}=40°C=2800h

Then:

Multiplier2\_0[7:0]=(8000h-4000h) / (4600h-2800h)=16384/7680=2.133

Convert the number 4.775 into our 8bit format: 00001000b (08h). BIOS should program Multiplier2\_0 register as 08h in this example.

Bits	Description						
7:0	<b>Multiplier2_0[7:0]</b> . Read-write. Reset: 0. BIOS: See above description. IF (PM2x00[TwoRampAlgorithmEn]==1) THEN It specifies the slope value of ramp1lo and ramp0lo. <table border="0" style="margin-left: 20px;"> <tr> <td style="padding-right: 20px;"><u>Bit</u></td> <td><u>Definition</u></td> </tr> <tr> <td>[1:0]</td> <td>Integer bits of the ramp slope</td> </tr> <tr> <td>[7:2]</td> <td>Fractional bits of the ramp slope</td> </tr> </table>	<u>Bit</u>	<u>Definition</u>	[1:0]	Integer bits of the ramp slope	[7:2]	Fractional bits of the ramp slope
<u>Bit</u>	<u>Definition</u>						
[1:0]	Integer bits of the ramp slope						
[7:2]	Fractional bits of the ramp slope						

**PM2x60 FanStatus**

Bits	Description
7:1	Reserved.
0	<b>Fan0SpeedTooSlow</b> . Read; write-1-to-clear. Reset: 0. 1=Fan0 runs slower than the value in the Fan0SpeedLimit{PM2x68, PM2x67}.

**PM2x61 FanINTRouteLo**

Bits	Description										
7:2	Reserved.										
1:0	<b>Fan0INTRoute</b> . Read-write. Reset: 0. <table border="0" style="margin-left: 20px;"> <tr> <td style="padding-right: 20px;"><u>Bits</u></td> <td><u>Definition</u></td> </tr> <tr> <td>00b</td> <td>No SCI/SMI generated</td> </tr> <tr> <td>01b</td> <td>SMI</td> </tr> <tr> <td>10b</td> <td>SMI or SCI depending on GEVENT13 routing</td> </tr> <tr> <td>11b</td> <td>No SCI/SMI generated</td> </tr> </table>	<u>Bits</u>	<u>Definition</u>	00b	No SCI/SMI generated	01b	SMI	10b	SMI or SCI depending on GEVENT13 routing	11b	No SCI/SMI generated
<u>Bits</u>	<u>Definition</u>										
00b	No SCI/SMI generated										
01b	SMI										
10b	SMI or SCI depending on GEVENT13 routing										
11b	No SCI/SMI generated										

**PM2x63 SampleFreqDiv**

Bits	Description
7:4	<b>LinearRangeOutLimit[4:1]</b> . Read-write. Reset:0. LinearRangeOutLimit[7:0] = {000b, LinearRangeOutLimit[4:1], 1b}. LinearRangeOutLimit specifies how close the fan duty follows the target duty cycle and is only used when the fan duty is changing. It is different from the PM2x0C [LinearRange0], which works like a hysteresis and used when fan duty is not changing.
3	Reserved.

2	<b>FanLinearEnhanceEn2</b> . Read-write. Reset: 0. 1=The positive hysteresis of fan duty is removed; <a href="#">PM2x0C [LinearRange0]</a> only applies to the negative direction; As a result, the fan duty increases once the temperature is increased instead of waiting for a hysteresis.										
1:0	<b>SampleFreqDiv</b> . Read-write. Reset: 0. This field specifies the sampling rate of Fan Speed. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Base (22.5 kHz)</td> </tr> <tr> <td>01b</td> <td>Base (22.5 kHz) / 2</td> </tr> <tr> <td>10b</td> <td>Base (22.5 kHz) / 4</td> </tr> <tr> <td>11b</td> <td>Base (22.5 kHz) / 8</td> </tr> </tbody> </table>	Bits	Definition	00b	Base (22.5 kHz)	01b	Base (22.5 kHz) / 2	10b	Base (22.5 kHz) / 4	11b	Base (22.5 kHz) / 8
Bits	Definition										
00b	Base (22.5 kHz)										
01b	Base (22.5 kHz) / 2										
10b	Base (22.5 kHz) / 4										
11b	Base (22.5 kHz) / 8										

**PM2x64 FanDebounceCounterLo**

Bits	Description
7:0	<b>FanDebounceCounter[7:0]</b> . Read-write. Reset: 0. Specifies the lower 8 bits of the debounced counter when measuring fan speed.

**PM2x65 FanDebounceCounterHi**

Bits	Description
7:0	<b>FanDebounceCounter[15:8]</b> . Read-write. Reset: 0. Specifies the high 8 bits of the debounced counter when measuring fan speed.

**PM2x66 Fan0DetectorControl**

Bits	Description
7:5	Reserved.
4	<b>ShutDownEnable</b> . Read-write. Reset: 0. 1=The system will be shut down if <a href="#">PM2x60[Fan0SpeedTooSlow]</a> remains 1 for more than 4 seconds.
3:2	Reserved.
1	<b>UseAverage</b> . Read-write. Reset: 0. 0=Do not to use average fan0 speed. 1=Use average fan0 speed.
0	<b>FanDetectorEnable</b> . Read-write. Reset: 0. 0=Disable fan0 speed measurement. 1=Enable fan0 speed measurement.

**PM2x67 Fan0SpeedLimitLo**

Bits	Description
7:0	<b>FanSpeedLimit[7:0]</b> . Read-write. Reset: 0. Specifies the lower 8 bits of Fan0SpeedLimit

**PM2x68 Fan0SpeedLimitHi**

Bits	Description
7:0	<b>FanSpeedLimit[15:8]</b> . Read-write. Reset: 0. Specifies the higher 8 bits of Fan0SpeedLimit.

**PM2x69 Fan0SpeedLo**

Bits	Description
7:0	<b>FanSpeed[7:0]</b> . Read-only; updated-by-hardware. Reset: 0. Specifies the lower 8 bits of fan0 speed.

**PM2x6A Fan0SpeedHi**

Bits	Description
7:0	<b>FanSpeed[15:8]</b> . Read-only; updated-by-hardware. Reset: 0. Specifies the higher 8 bits of fan0 speed.

**PM2x8A TempTsiLo**

Bits	Description
7:0	<b>TempTsi[7:0]</b> . IF (PM2x8F [TempTsiWe]==1) THEN Read-Write. ELSE Read-only; updated-by-hardware. ENDIF. Reset: 0. Specifies the lower 8 bits of TempTsi.

**PM2x8B TempTsiHi**

Bits	Description
7:0	<b>TempTsi[15:8]</b> . IF (PM2x8F [TempTsiWe]==1) THEN Read-Write. ELSE Read-only; updated-by-hardware. ENDIF. Reset: 0. Specifies the higher 8 bits of TempTsi.

**PM2x8C TempTsiLimitLo**

Bits	Description
7:0	<b>TempTsiLimit[7:0]</b> . Read-write. Reset: 0. Specifies the lower 8 bits of TempTsiLimit.

**PM2x8D TempTsiLimitHi**

Bits	Description
7:0	<b>TempTsiLimit[15:8]</b> . Read-write. Reset: 0. Specifies the higher 8 bits of TempTsiLimit.

**PM2x8E TempTsiChangeLimit**

If (TempTsiChangeLimit != 0), filtering is applied to TempTsi{PM2x8B [TempTsiHi], PM2x8A [TempTsiLo]} as below:

- If TempTsi\_New > (TempTsi\_Old + (TempTsiChangeLimit<<6)), then
- If TempTsi\_New < (TempTsi\_Old - (TempTsiChangeLimit<<6)), then
- If (Temp\_Old - (TempTsiChangLimit<<6)) <= TempTsi\_New <= (TempTsi\_Old + (TempChangeLimit<<6)), then

Bits	Description
7:0	<b>TempTsiChangeLimit.</b> Read-write. Reset: 0.

**PM2x8F TempTsiWe**

Bits	Description
7:6	Reserved.
5	<b>TempTsiWe: TempTsi Write Enable.</b> Read-write. Reset: 0. 0=TempTsi registers, <a href="#">PM2x8B [TempTsiHi]</a> and <a href="#">PM2x8A [TempTsiLo]</a> , are read-only and updated by hardware with the result from TempTsi sensor. 1=TempTsi registers are writable only by host and IMC; they are not updated by hardware with the result from the TempTsi sensor.
4:0	Reserved.

**PM2x90 TempTsiStatus**

Bits	Description
7:6	Reserved.
5	<b>TempTsiStatus.</b> Write-1-to-clear; Read. Reset: 0. 1=TempTsi{ <a href="#">PM2x8B [TempTsiHi]</a> , <a href="#">PM2x8A [TempTsiLo]</a> } is out of the limit.
4:0	Reserved.

**PM2x92 TempTsiControl**

Bits	Description										
7:4	Reserved.										
3:2	<b>TempTsiControl.</b> Read-write. Reset: 0. TempTsi sensor is enabled if (TempTsiControl != 0). <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Disable.</td> </tr> <tr> <td>01b</td> <td>If (TempTsi &gt; TempTsiLimit), then <a href="#">PM2x90[TempTsiStatus]</a> = 1.</td> </tr> <tr> <td>10b</td> <td>If (TempTsi &lt; TempTsiLimit), then <a href="#">PM2x90[TempTsiStatus]</a> = 1.</td> </tr> <tr> <td>11b</td> <td>If ((TempTsiHi &gt; TempTsiLimitLo)    (TempTsiHi &lt; TempTsiLimitHi)), then <a href="#">PM2x90[TempTsiStatus]</a> = 1.</td> </tr> </tbody> </table>	Bits	Definition	00b	Disable.	01b	If (TempTsi > TempTsiLimit), then <a href="#">PM2x90[TempTsiStatus]</a> = 1.	10b	If (TempTsi < TempTsiLimit), then <a href="#">PM2x90[TempTsiStatus]</a> = 1.	11b	If ((TempTsiHi > TempTsiLimitLo)    (TempTsiHi < TempTsiLimitHi)), then <a href="#">PM2x90[TempTsiStatus]</a> = 1.
Bits	Definition										
00b	Disable.										
01b	If (TempTsi > TempTsiLimit), then <a href="#">PM2x90[TempTsiStatus]</a> = 1.										
10b	If (TempTsi < TempTsiLimit), then <a href="#">PM2x90[TempTsiStatus]</a> = 1.										
11b	If ((TempTsiHi > TempTsiLimitLo)    (TempTsiHi < TempTsiLimitHi)), then <a href="#">PM2x90[TempTsiStatus]</a> = 1.										
1:0	Reserved.										

**PM2x94 TempTsiINTRoute**

Bits	Description
7:4	Reserved.

3:2	<b>TempTsiINTRoute.</b> Read-write. Reset: 0. <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>No SCI/SMI generated.</td> </tr> <tr> <td>01b</td> <td>SMI</td> </tr> <tr> <td>10b</td> <td>SMI or SCI according to GEVENT 13 INT routing.</td> </tr> <tr> <td>11b</td> <td>No SCI/SMI generated.</td> </tr> </tbody> </table>	Bits	Definition	00b	No SCI/SMI generated.	01b	SMI	10b	SMI or SCI according to GEVENT 13 INT routing.	11b	No SCI/SMI generated.
Bits	Definition										
00b	No SCI/SMI generated.										
01b	SMI										
10b	SMI or SCI according to GEVENT 13 INT routing.										
11b	No SCI/SMI generated.										
1:0	Reserved.										

**PM2xDF TempTsiRstSel**

Bits	Description
7:6	Reserved.
5	<b>TempTsiRstSel.</b> Read-write. Reset: 0. 1=Thermal diode monitoring function is not stopped by reset.
4:0	Reserved.

**PM2xE0 AlertThermaltripStatus**

Bits	Description
7:2	Reserved.
1	<b>ThermalTripStatus.</b> Read-only; updated-by-hardware. Reset: 0. 0=Current temperature is not above ThermalTripLimit. 1=Current temperature is above ThermalTripLimit.
0	<b>AlertStatus.</b> Read-only; updated-by-hardware. Reset: 0. 0=Current temperature is not above AlertLimit. 1=Current temperature is above AlertLimit.

**PM2xE1 AlertLimitLo**

Bits	Description
7:0	<b>AlertLimit[7:0].</b> Read-write. Reset: 0.

**PM2xE2 AlertLimitHi**

Bits	Description
7:0	<b>AlertLimit[15:8].</b> Read-write. Reset: 0.

**PM2xE3 ThermalTripLimitLo**

Bits	Description
7:0	<b>ThermalTripLimit[7:0].</b> Read-write. Reset: 0.

**PM2xE4 ThermalTripLimitHi**

Bits	Description
7:0	<b>ThermalTripLimit[15:8]</b> . Read-write. Reset: 0.

**PM2xE5 AlertThermaltripControl**

Bits	Description								
7:5	<b>TempSelAlert</b> . Read-write. Reset: 0. This field selects which temperature sensor is the event source. It converts the selected Temp input pin into either TAlert or ThermalTrip function. <table border="1"> <thead> <tr> <th>Bits</th> <th>Field Definition</th> </tr> </thead> <tbody> <tr> <td>100b-000b</td> <td>Reserved</td> </tr> <tr> <td>101b</td> <td>TempTsi</td> </tr> <tr> <td>111b-110b</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Field Definition	100b-000b	Reserved	101b	TempTsi	111b-110b	Reserved
Bits	Field Definition								
100b-000b	Reserved								
101b	TempTsi								
111b-110b	Reserved								
4:2	Reserved.								
1:0	<b>AlertControl</b> . Read-write. Reset: 0. <table border="1"> <thead> <tr> <th>Bit</th> <th>Field Definition</th> </tr> </thead> <tbody> <tr> <td>[1]</td> <td>Enable ThermalTrip on the selected Temp input</td> </tr> <tr> <td>[0]</td> <td>Enable TAlert on the selected Temp input</td> </tr> </tbody> </table>	Bit	Field Definition	[1]	Enable ThermalTrip on the selected Temp input	[0]	Enable TAlert on the selected Temp input		
Bit	Field Definition								
[1]	Enable ThermalTrip on the selected Temp input								
[0]	Enable TAlert on the selected Temp input								

**3.25.15 Standard ACPI Registers**

These are the standard registers defined by the ACPI specification. In order to access these ACPI registers, [PMx74\[DecEnAcpi\]](#) must be programmed to 1. ACPI registers are accessed through two methods:

- IO mapped access through base addresses defined by [PMx60](#) through [PMx6E](#).
- Memory mapped or IO mapped access through the AcpiMmio region. The ACPI registers range from [PMx24\[AcpiMmioAddr\]+800h](#) to [PMx24\[AcpiMmioAddr\]+8FFh](#). See [PMx24 \[AcpiMmioEn\]](#).

**3.25.15.1 AcpiPmEvtBlk**

The IO mapped base address of this register block is defined by [PMx60](#).

**AcpiPmEvtBlkx00 Pm1Status**

Bits	Description
15	<b>WakeStatus</b> . Read; set-by-hardware; write-1-to-clear. Reset: 0. 1=The system is in the sleep state and a wake-up event occurs.
14	<b>PciExpWakeStatus</b> . Read; set-by-hardware; write-1-to-clear. Reset: 0. 1=The system wake is due to a PCI Express <sup>®</sup> wakeup event.
13:11	Reserved.
10	<b>RtcStatus</b> . Read; set-by-hardware; write-1-to-clear. Reset: 0. 1=RTC generates an alarm. If both <a href="#">AcpiPmEvtBlkx02[RtcEn]</a> and this bit are set to 1, a power management event: SCI, SMI, or resume event, is generated.
9	Reserved.

8	<b>PwrBtnStatus.</b> Read; set-by-hardware; write-1-to-clear. Reset: 0. Power button status bit. 1=The Power Button is pressed. In the system working state, if <a href="#">AcpiPmEvtBlkx02[PwrBtnEn]</a> and this bit are both set to 1, an interrupt event is raised. In the sleeping or soft-off state, a wake event is generated when the power button is pressed regardless of the setting of <a href="#">AcpiPmEvtBlkx02[PwrBtnEn]</a> .
7:6	Reserved.
5	<b>GblStatus.</b> Read; set-by-hardware; write-1-to-clear. Reset: 0. 1=An SCI is generated due to the BIOS wanting the attention of the SCI handler; Writing 1 to <a href="#">PMx74[BiosRIs]</a> sets this bit.
4	<b>BmStatus.</b> Read; set-by-hardware; write-1-to-clear. Reset: 0.
3:1	Reserved.
0	<b>TmrStatus.</b> Read; set-by-hardware; write-1-to-clear. Reset: 0. Timer carry status bit. 1=The 31st bit of the 32-bit counter changes from low to high or high to low. If both <a href="#">AcpiPmEvtBlkx02[TmrEn]</a> and this bit are set to 1, an interrupt event is raised.

### AcpiPmEvtBlkx02 Pm1Enable

Bits	Description
15	Reserved.
14	<b>PciExpWakeDis.</b> Read-write. Reset: 1. 1=Disable the inputs to <a href="#">AcpiPmEvtBlkx00[PciExpWakeStatus]</a> from waking the system.
13:11	Reserved.
10	<b>RtcEn.</b> Read-write. Reset: 0. 1=A wake event is generated whenever <a href="#">AcpiPmEvtBlkx00[RtcStatus]</a> is also set to 1.
9	Reserved.
8	<b>PwrBtnEn.</b> Read-write. Reset: 0. 1=A power management event (SCI or wake) is generated whenever <a href="#">AcpiPmEvtBlkx00[PwrBtnStatus]</a> is also set to 1.
7:6	Reserved.
5	<b>GblEn.</b> Read-write. Reset: 0. 1=An SCI is raised whenever <a href="#">AcpiPmEvtBlkx00[GblStatus]</a> is also set to 1.
4:1	Reserved.
0	<b>TmrEn.</b> Read-write. Reset: 0. This is the timer carry interrupt enable bit. 1=An SCI event is generated whenever the <a href="#">AcpiPmEvtBlkx00[TmrStatus]</a> is set to 1. 0=No interrupt is generated when the <a href="#">AcpiPmEvtBlkx00[TmrStatus]</a> bit is set to 1.

### 3.25.15.2 AcpiPm1CntBlk

The IO mapped base address of this register block is defined by [PMx62](#).

### AcpiPm1CntBlkx00 PmControl

Bits	Description
15:14	Reserved.
13	<b>SlpTypeEn.</b> RAZ; write-1-only. Reset: 0. 1=The system sequences into the sleeping state defined by <a href="#">SlpTyp</a> when <a href="#">PMxBE [SlpTypeControl]</a> is set to 1.



12:10	<b>SlpTyp.</b> Read-write. Reset: 0. Specifies the sleep state the system enters if SlpTypeEn is set to 1. This design currently implements 5 states: S0, S1, S3, S4, and S5.
9:3	Reserved.
2	<b>GblRls.</b> RAZ; write-only. Reset: 0. If <b>SMIxB0</b> [17:16] is set to 01b, writing 1 to this bit generates SMI and sets <b>SMIx88</b> [8].
1	<b>BmRld.</b> Read-write. Reset: 0.
0	<b>SciEn.</b> Read-write. Reset: 0. Selects the power management event to be either an SCI or SMI interrupt for the power management events. 1=Power management events generate SCI interrupts; 0=Power management events generate SMI interrupts.

### 3.25.15.3 AcpiPm2CntBlk

The IO mapped base address of this register block is defined by **PMx6E**.

#### AcpiPm2CntBlkx00 Pm2Control

Bits	Description
7:1	Reserved.
0	<b>ArbDis.</b> Read-write. Reset: 0.

### 3.25.15.4 AcpiPmTmrBlk

The IO mapped base address of this register block is defined by **PMx64**.

#### AcpiPmTmrBlkx00 TmrValue/ETmrValue

Bits	Description
31:0	<b>TmrValue.</b> Read-only. Reset: X. It returns the running count of the power management timer (ACPI timer).

### 3.25.15.5 CpuCntBlk

The IO mapped base address of this register block is defined by **PMx66**.

#### CpuCntBlkx00 ClkValue

Bits	Description																				
31:5	Reserved.																				
4	<b>ThtEn.</b> Read-write. Reset: 0. 1=Enable clock throttling as programmed in ClkValue.																				
3:1	<b>ClkValue.</b> Read-write. Reset: 0. Specifies the throttle interval for STPCLK in software clock throttling. <table border="1" data-bbox="267 1732 1128 1900"> <thead> <tr> <th>Bits</th> <th>Definition</th> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>50%</td> <td>100b</td> <td>50%</td> </tr> <tr> <td>001b</td> <td>12.5%</td> <td>101b</td> <td>62.5%</td> </tr> <tr> <td>010b</td> <td>25%</td> <td>110b</td> <td>75%</td> </tr> <tr> <td>011b</td> <td>37.5%</td> <td>111b</td> <td>87.5%</td> </tr> </tbody> </table>	Bits	Definition	Bits	Definition	000b	50%	100b	50%	001b	12.5%	101b	62.5%	010b	25%	110b	75%	011b	37.5%	111b	87.5%
Bits	Definition	Bits	Definition																		
000b	50%	100b	50%																		
001b	12.5%	101b	62.5%																		
010b	25%	110b	75%																		
011b	37.5%	111b	87.5%																		
0	Reserved.																				

**CpuCntBlkx04 PLv12**

Bits	Description
7:0	<b>PLv12</b> . RAZ; read-only. Reset: 0. Reads to this register return all zeros and generate an “enter C2” sequence to APU.

**CpuCntBlkx05 PLv13**

Bits	Description
7:0	<b>PLv13</b> . RAZ; read-only. Reset: 0. Reads to this register return all zeros and generate an “enter C3” sequence to APU.

**3.25.15.6 AcpiGpe0Blk**

The IO mapped base address of this register block is defined by [PMx68](#).

**AcpiGpe0Blkx00 EventStatus**

Bits	Description
31:0	<b>EventStatus</b> . Read; write-1-to-clear. Reset: X. Each bit represents an ACPI event status. For each bit, 1=The selected event input equals to the corresponding value in <a href="#">SMIx08 [SciTrig]</a> . Configuration for the events are located at <a href="#">SMIx08 [SciTrig]</a> through <a href="#">SMIx78 [SciMap14]</a> . The status bits are also mirrored in <a href="#">SMIx00 [Event_Status]</a> .

**AcpiGpe0Blkx04 EventEnable**

Bits	Description
31:0	<b>EventEnable</b> . Read-write. Reset: 0. Each bit controls whether ACPI should generate wakeup and SCI interrupt. The enable bits are also mirrored in <a href="#">SMIx04 [Event_Enable]</a> .

**3.25.15.7 SmiCmdBlk**

The IO mapped base address of this register block is defined by [PMx6A](#).

**SmiCmdBlkx00 SmiCmdPort**

Bits	Description
7:0	<b>SmiCmdPort</b> . Read-write. Reset: 0. When SMI command port is enabled, a write to this port generates SMI, NMI or IRQ3 depending on the setting of <a href="#">SMIxB0[23:22]</a> . A read from this port returns the previously written value but does not generate SMI.

**SmiCmdBlkx01 SmiCmdStatus**

Bits	Description
7:0	<b>SmiCmdStatus</b> . Read-write. Reset: 0. Used by BIOS and OS.

### 3.25.16 SMI Registers

SMI register space is accessed through the AcpiMmio region, which can be memory mapped or IO mapped. The SMI registers range from [PMx24\[AcpiMmioAddr\]+200h](#) to [PMx24\[AcpiMmioAddr\]+2FFh](#). See [PMx24\[AcpiMmioEn\]](#).

#### SMIx00 Event\_Status

Bits	Description
31:0	<b>EventStatus.</b> Read; write-1-to-clear. Reset: X. This is a mirror register of the standard ACPI register <a href="#">AcpiGpe0Blkx00 [EventStatus]</a> . Each event status bit is set when the selected event input equals to the corresponding value in <a href="#">SMIx08 [SciTrig]</a> .

#### SMIx04 Event\_Enable

Bits	Description
31:0	<b>EventEnable.</b> Read-write. Reset: 0. This is the mirror register of the standard ACPI register <a href="#">AcpiGpe0Blkx04 [EventEnable]</a> . Each bit controls whether ACPI should generate wake up and SCI interrupt.

#### SMIx08 SciTrig

Each bit in this register controls the way to set each corresponding bit in [SMIx00 \[Event\\_Status\]](#).

Bits	Description
31	<b>SciTrig31.</b> Read-write. Reset: 1. The bit controls the way to set Event_Status bit 31. 0=Active low. 1=Active high.
30	<b>SciTrig30.</b> Read-write. Reset: 1. The bit controls the way to set Event_Status bit 30. 0=Active low. 1=Active high.
29	<b>SciTrig29.</b> Read-write. Reset: 1. The bit controls the way to set Event_Status bit 29. 0=Active low. 1=Active high.
28	<b>SciTrig28.</b> Read-write. Reset: 1. The bit controls the way to set Event_Status bit 28. 0=Active low. 1=Active high.
27	<b>SciTrig27.</b> Read-write. Reset: 1. The bit controls the way to set Event_Status bit 27. 0=Active low. 1=Active high.
26	<b>SciTrig26.</b> Read-write. Reset: 1. The bit controls the way to set Event_Status bit 26. 0=Active low. 1=Active high.
25	<b>SciTrig25.</b> Read-write. Reset: 1. The bit controls the way to set Event_Status bit 25. 0=Active low. 1=Active high.
24	<b>SciTrig24.</b> Read-write. Reset: 1. The bit controls the way to set Event_Status bit 24. 0=Active low. 1=Active high.
23	<b>SciTrig23.</b> Read-write. Reset: 1. The bit controls the way to set Event_Status bit 23. 0=Active low. 1=Active high.
22	<b>SciTrig22.</b> Read-write. Reset: 1. The bit controls the way to set Event_Status bit 22. 0=Active low. 1=Active high.

21	<b>SciTrig21.</b> Read-write. Reset: 1. The bit controls the way to set Event_Status bit 21. 0=Active low. 1=Active high.
20	<b>SciTrig20.</b> Read-write. Reset: 1. The bit controls the way to set Event_Status bit 20. 0=Active low. 1=Active high.
19	<b>SciTrig19.</b> Read-write. Reset: 1. The bit controls the way to set Event_Status bit 19. 0=Active low. 1=Active high.
18	<b>SciTrig18.</b> Read-write. Reset: 1. The bit controls the way to set Event_Status bit 18. 0=Active low. 1=Active high.
17	<b>SciTrig17.</b> Read-write. Reset: 1. The bit controls the way to set Event_Status bit 17. 0=Active low. 1=Active high.
16	<b>SciTrig16.</b> Read-write. Reset: 1. The bit controls the way to set Event_Status bit 16. 0=Active low. 1=Active high.
15	<b>SciTrig15.</b> Read-write. Reset: 1. The bit controls the way to set Event_Status bit 15. 0=Active low. 1=Active high.
14	<b>SciTrig14.</b> Read-write. Reset: 1. The bit controls the way to set Event_Status bit 14. 0=Active low. 1=Active high.
13	<b>SciTrig13.</b> Read-write. Reset: 1. The bit controls the way to set Event_Status bit 13. 0=Active low. 1=Active high.
12	<b>SciTrig12.</b> Read-write. Reset: 1. The bit controls the way to set Event_Status bit 12. 0=Active low. 1=Active high.
11	<b>SciTrig11.</b> Read-write. Reset: 1. The bit controls the way to set Event_Status bit 11. 0=Falling edge. 1=Active high.
10	<b>SciTrig10.</b> Read-write. Reset: 1. The bit controls the way to set Event_Status bit 10. 0=Active low. 1=Active high.
9	<b>SciTrig9.</b> Read-write. Reset: 1. The bit controls the way to set Event_Status bit 9. 0=Active low. 1=Active high.
8	<b>SciTrig8.</b> Read-write. Reset: 1. The bit controls the way to set Event_Status bit 8. 0=Active low. 1=Active high.
7	<b>SciTrig7.</b> Read-write. Reset: 1. The bit controls the way to set Event_Status bit 7. 0=Active low. 1=Active high.
6	<b>SciTrig6.</b> Read-write. Reset: 1. The bit controls the way to set Event_Status bit 6. 0=Active low. 1=Active high.
5	<b>SciTrig5.</b> Read-write. Reset: 1. The bit controls the way to set Event_Status bit 5. 0=Active low. 1=Active high.
4	<b>SciTrig4.</b> Read-write. Reset: 1. The bit controls the way to set Event_Status bit 4. 0=Active low. 1=Active high.
3	<b>SciTrig3.</b> Read-write. Reset: 1. The bit controls the way to set Event_Status bit 3. 0=Active low. 1=Active high.
2	<b>SciTrig2.</b> Read-write. Reset: 1. The bit controls the way to set Event_Status bit 2. 0=Active low. 1=Active high.
1	<b>SciTrig1.</b> Read-write. Reset: 1. The bit controls the way to set Event_Status bit 1. 0=Active low. 1=Active high.
0	<b>SciTrig0.</b> Read-write. Reset: 1. The bit controls the way to set Event_Status bit 0. 0=Active low. 1=Active high.

**SMIx0C SciLevel**

Reset: 0000\_0000h. This register specifies the trigger mode for each of the corresponding bit in [SMIx00 \[Event\\_Status\]](#).

Bits	Description
31	<b>SciLevel31.</b> Read-write. 0=Edge trigger. 1=Level trigger.
30	<b>SciLevel30.</b> Read-write. 0=Edge trigger. 1=Level trigger.
29	<b>SciLevel29.</b> Read-write. 0=Edge trigger. 1=Level trigger.
28	<b>SciLevel28.</b> Read-write. 0=Edge trigger. 1=Level trigger.
27	<b>SciLevel27.</b> Read-write. 0=Edge trigger. 1=Level trigger.
26	<b>SciLevel26.</b> Read-write. 0=Edge trigger. 1=Level trigger.
25	<b>SciLevel25.</b> Read-write. 0=Edge trigger. 1=Level trigger.
24	<b>SciLevel24.</b> Read-write. 0=Edge trigger. 1=Level trigger.
23	<b>SciLevel23.</b> Read-write. 0=Edge trigger. 1=Level trigger.
22	<b>SciLevel22.</b> Read-write. 0=Edge trigger. 1=Level trigger.
21	<b>SciLevel21.</b> Read-write. 0=Edge trigger. 1=Level trigger.
20	<b>SciLevel20.</b> Read-write. 0=Edge trigger. 1=Level trigger.
19	<b>SciLevel19.</b> Read-write. 0=Edge trigger. 1=Level trigger.
18	<b>SciLevel18.</b> Read-write. 0=Edge trigger. 1=Level trigger.
17	<b>SciLevel17.</b> Read-write. 0=Edge trigger. 1=Level trigger.
16	<b>SciLevel16.</b> Read-write. 0=Edge trigger. 1=Level trigger.
15	<b>SciLevel15.</b> Read-write. 0=Edge trigger. 1=Level trigger.
14	<b>SciLevel14.</b> Read-write. 0=Edge trigger. 1=Level trigger.
13	<b>SciLevel13.</b> Read-write. 0=Edge trigger. 1=Level trigger.
12	<b>SciLevel12.</b> Read-write. 0=Edge trigger. 1=Level trigger.
11	<b>SciLevel11.</b> Read-write. 0=Edge trigger. 1=Level trigger.
10	<b>SciLevel10.</b> Read-write. 0=Edge trigger. 1=Level trigger.
9	<b>SciLevel9.</b> Read-write. 0=Edge trigger. 1=Level trigger.
8	<b>SciLevel8.</b> Read-write. 0=Edge trigger. 1=Level trigger.
7	<b>SciLevel7.</b> Read-write. 0=Edge trigger. 1=Level trigger.
6	<b>SciLevel6.</b> Read-write. 0=Edge trigger. 1=Level trigger.
5	<b>SciLevel5.</b> Read-write. 0=Edge trigger. 1=Level trigger.
4	<b>SciLevel4.</b> Read-write. 0=Edge trigger. 1=Level trigger.
3	<b>SciLevel3.</b> Read-write. 0=Edge trigger. 1=Level trigger.
2	<b>SciLevel2.</b> Read-write. 0=Edge trigger. 1=Level trigger.
1	<b>SciLevel1.</b> Read-write. 0=Edge trigger. 1=Level trigger.
0	<b>SciLevel0.</b> Read-write. 0=Edge trigger. 1=Level trigger.

**SMIx10 SmiSciStatus**

Bits	Description
31:0	<b>SmiSciStatus.</b> Read; write-1-to-clear. Reset: 0. Each bit indicates the corresponding SmiSci status. The input of each bit is controlled by the corresponding <a href="#">SMIx08 [SciTrig]</a> bit. Note: this function can be considered as a superset of <a href="#">SMIx00 [Event_Status]</a> . When one of the bits is set and its <a href="#">SMIx14 [SmiSciEn]</a> is also set, it triggers a SMI to call the BIOS. After the BIOS has serviced the SMM and cleared its status, the internal logic automatically sets the corresponding <a href="#">SMIx00 [Event_Status]</a> bit and thereby triggering a SCI.

**SMIx14 SmiSciEn**

Bits	Description
31:0	<b>SmiSciEn.</b> Read-write. Reset: 0. Each bit controls if SMI message is generated when the corresponding <a href="#">SMIx10 [SmiSciStatus]</a> bit is set to 1. For each bit, 0=Not to send SMI message when the corresponding <a href="#">SMIx10 [SmiSciStatus]</a> bit is set. 1=Send SMI message when the corresponding <a href="#">SMIx10 [SmiSciStatus]</a> bit is set.

**SMIx18 SwSciEn**

Bits	Description
31:0	<b>SwSciEn.</b> Read-write. Reset: 0. This register is used as a software mechanism to trigger SCI. For each bit, 1=Software can write to <a href="#">SMIx1C [SwSciData]</a> and set the corresponding <a href="#">SMIx00 [Event_Status]</a> bit. Note: the setting of this bit needs to match with <a href="#">SMIx08 [SciTrig]</a> and <a href="#">SMIx0C [SciLevl]</a> in order to set the status bit.

**SMIx1C SwSciData**

Bits	Description
31:0	<b>SwSciData.</b> Read-write. Reset: 0. This is the software data path to set the corresponding <a href="#">SMIx00 [Event_Status]</a> bit when <a href="#">SMIx18 [SwSciEn]</a> is set.

**SMIx20 SciSleepDisable**

Bits	Description
31:0	<b>SciSleepDisable.</b> Read-write. Reset: 0. This register is used to ignore EVENT pins that are powered in the main power domain instead of auxiliary power domain. For each bit, 1=The corresponding <a href="#">SMIx00 [Event_Status]</a> bit is masked off whenever the system goes to S3 or higher sleep state.

**SMIx30 CapturedData**

Bits	Description
31:0	<b>CapturedData</b> . Read-only. Reset: X. This is the buffer to capture write data for the last transaction that caused an SMI. Note: this buffer has no meaning for a read trap.

**SMIx34 CapturedValid**

Bits	Description										
7:4	Reserved.										
3:0	<b>CapturedValid</b> . Read-only. Reset: X. This is the byte valid buffer to signal which byte is captured for the last transaction that caused the SMI. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Bit</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>[0]</td> <td>byte 0</td> </tr> <tr> <td>[1]</td> <td>byte 1</td> </tr> <tr> <td>[2]</td> <td>byte 2</td> </tr> <tr> <td>[3]</td> <td>byte 3</td> </tr> </tbody> </table>	Bit	Definition	[0]	byte 0	[1]	byte 1	[2]	byte 2	[3]	byte 3
Bit	Definition										
[0]	byte 0										
[1]	byte 1										
[2]	byte 2										
[3]	byte 3										

**SMIx38 EPBIF\_AER\_Straps**

Bits	Description
31:28	Reserved.
27	<b>StrapBifInternalErrEnFch</b> . Read-write. Reset: 0. Internal error enable.
26	<b>StrapBifPoisonedAdvisoryNonfatalAFch</b> . Read-write. Reset: 1. Poisoned TLP as advisory nonfatal.
25	<b>StrapBifAcsDirectTranslatedP2pFch</b> . Read-write. Reset: 1. ACS direct translated P2P enable.
24	<b>StrapBifAcsUpstreamForwardingFch</b> . Read-write. Reset: 1. ACS upstream forwarding enable.
23	<b>StrapBifAcsP2pCompletionRedirectFch</b> . Read-write. Reset: 1. ACS P2P completion redirect enable.
22	<b>StrapBifAcsP2pRequestRedirectFch</b> . Read-write. Reset: 1. ACS P2P request redirect enable.
21	<b>StrapBifAcsTranslationBlockingFch</b> . Read-write. Reset: 1. ACS translation blocking enable.
20	<b>StrapBifAcsSourceValidationFch</b> . Read-write. Reset: 1. ACS source validation enable.
19	<b>StrapBifAcsEnFch</b> . Read-write. Reset: 1. ACS enable.
18	<b>StrapBifFirstRcvdErrLogFch</b> . Read-write. Reset: 1. First received error log.
17	<b>StrapBifEcrcCheckEnFch</b> . Read-write. Reset: 1. ECRC check enable.
16	<b>StrapBifEcrcGenEnFch</b> . Read-write. Reset: 1. ECRC generate enable.
15	<b>StrapBifCplAbortErrEnFch</b> . Read-write. Reset: 1. Completer abort error enable.
14	<b>StrapBifRxIgnoreVend0UrFch</b> . Read-write. Reset: 1. Ignore Vendor 0 error.
13	<b>StrapBifRxIgnoreTcErrFch</b> . Read-write. Reset: 1. Ignore traffic class error.
12	<b>StrapBifRxIgnoreMsgErrFch</b> . Read-write. Reset: 1. Ignore message error.
11	<b>StrapBifRxIgnoreMaxPayloadErrFch</b> . Read-write. Reset: 1. Ignore maximum payload error.

10	<b>StrapBifRxIgnoreLenMismatchErrFch.</b> Read-write. Reset: 1. Ignore length mismatch error.
9	<b>StrapBifRxIgnoreIoUrErrFch.</b> Read-write. Reset: 1. Ignore IO UR error.
8	<b>StrapBifRxIgnoreIoErrFch.</b> Read-write. Reset: 1. Ignore IO error.
7	<b>StrapBifRxIgnoreEpErrFch.</b> Read-write. Reset: 1. Ignore poisoned TLP error.
6	<b>StrapBifRxIgnoreCplErrFch.</b> Read-write. Reset: 1. Ignore completion error.
5	<b>StrapBifRxIgnoreCfgUrFch.</b> Read-write. Reset: 1. Ignore configuration UR error.
4	<b>StrapBifRxIgnoreCfgErrFch.</b> Read-write. Reset: 1. Ignore configuration error.
3	<b>StrapBifRxIgnoreBeErrFch.</b> Read-write. Reset: 1. Ignore byte enable error.
2	<b>StrapBifErrReportingDisFch.</b> Read-write. Reset: 1. Error reporting disable.
1	<b>StrapBifAerEnFch.</b> Read-write. Reset: 1. AER enable.
0	<b>StrapBifStickyOverrideS5.</b> Read-write. Reset: 0. 1=Values in this register would override straps loaded from EEPROM.

### SMI<sub>x</sub>3C DataErrorStatus

Bits	Description
31:8	Reserved.
7	<b>SIrqIochk.</b> Read; write-1-to-clear. Reset: 0. Serial Iochk error.
6	<b>SataPerr.</b> Read; write-1-to-clear. Reset: 0. BIOS: See 2.15.3.9. SATA controller internal parity error status.
5	<b>UmiUncorrectableErr.</b> Read; write-1-to-clear. Reset: 0. UMI uncorrectable error status.
4	<b>UmiCorrectableErr.</b> Read; write-1-to-clear. Reset: 0. UMICorrectable error status.
3	<b>AbUmiGppPerr.</b> Read; write-1-to-clear. Reset: 0. AB/UMI/GPP parity error status.
2	<b>ApuGppSerr.</b> Read; write-1-to-clear. Reset: 0. APU/GPP devices SERR error status.
1	<b>InternalPerr.</b> Read; write-1-to-clear. Reset: 0. Internal devices PERR error status.
0	<b>InternalSerr.</b> Read; write-1-to-clear. Reset: 0. Internal devices SERR error status.

### SMI<sub>x</sub>40 SciMap0

Bits	Description
31:29	Reserved.
28:24	<b>SciMap3.</b> Read-write. Reset: 0. Specifies the mapping of GEVENT3 to one of the 32 bits in <a href="#">SMI<sub>x</sub>00 [Event_Status]</a> . See: SciMap0.
23:21	Reserved.
20:16	<b>SciMap2.</b> Read-write. Reset: 0. Specifies the mapping of GEVENT2 to one of the 32 bits in <a href="#">SMI<sub>x</sub>00 [Event_Status]</a> . See: SciMap0.
15:13	Reserved.
12:8	<b>SciMap1.</b> Read-write. Reset: 0. Specifies the mapping of GEVENT1 to one of the 32 bits in <a href="#">SMI<sub>x</sub>00 [Event_Status]</a> . See: SciMap0.



7:5	Reserved.										
4:0	<p><b>SciMap0.</b> Read-write. Reset: 0. Specifies the mapping of GEVENT0 to one of the 32 bits in <a href="#">SMIx00 [Event_Status]</a>.</p> <table border="0"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>00000b</td> <td>map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [0]</td> </tr> <tr> <td>00001b</td> <td>map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [1]</td> </tr> <tr> <td>...</td> <td>...</td> </tr> <tr> <td>11111b</td> <td>map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [31]</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	00000b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [0]	00001b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [1]	...	...	11111b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [31]
<u>Bits</u>	<u>Definition</u>										
00000b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [0]										
00001b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [1]										
...	...										
11111b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [31]										

### SMIx44 SciMap1

Bits	Description										
31:29	Reserved.										
28:24	<b>SciMap7.</b> Read-write. Reset: 0. Specifies the mapping of GEVENT7 to one of the 32 bits in <a href="#">SMIx00 [Event_Status]</a> . See: SciMap4.										
23:21	Reserved.										
20:16	<b>SciMap6.</b> Read-write. Reset: 0. Specifies the mapping of GEVENT6 to one of the 32 bits in <a href="#">SMIx00 [Event_Status]</a> . See: SciMap4.										
15:13	Reserved.										
12:8	<b>SciMap5.</b> Read-write. Reset: 0. Specifies the mapping of GEVENT5 to one of the 32 bits in <a href="#">SMIx00 [Event_Status]</a> . See: SciMap4.										
7:5	Reserved.										
4:0	<p><b>SciMap4.</b> Read-write. Reset: 0. Specifies the mapping of GEVENT4 to one of the 32 bits in <a href="#">SMIx00 [Event_Status]</a>.</p> <table border="0"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>00000b</td> <td>map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [0]</td> </tr> <tr> <td>00001b</td> <td>map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [1]</td> </tr> <tr> <td>...</td> <td>...</td> </tr> <tr> <td>11111b</td> <td>map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [31]</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	00000b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [0]	00001b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [1]	...	...	11111b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [31]
<u>Bits</u>	<u>Definition</u>										
00000b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [0]										
00001b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [1]										
...	...										
11111b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [31]										

### SMIx48 SciMap2

Bits	Description
31:29	Reserved.
28:24	<b>SciMap11.</b> Read-write. Reset: 0. Specifies the mapping of GEVENT11 to one of the 32 bits in <a href="#">SMIx00 [Event_Status]</a> . See: SciMap8.
23:21	Reserved.
20:16	<b>SciMap10.</b> Read-write. Reset: 0. Specifies the mapping of GEVENT10 to one of the 32 bits in <a href="#">SMIx00 [Event_Status]</a> . See: SciMap8.
15:13	Reserved.
12:8	<b>SciMap9.</b> Read-write. Reset: 0. Specifies the mapping of GEVENT9 to one of the 32 bits in <a href="#">SMIx00 [Event_Status]</a> . See: SciMap8.

7:5	Reserved.										
4:0	<p><b>SciMap8.</b> Read-write. Reset: 0. Specifies the mapping of GEVENT8 to one of the 32 bits in <a href="#">SMIx00 [Event_Status]</a>.</p> <table border="0"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>00000b</td> <td>map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [0]</td> </tr> <tr> <td>00001b</td> <td>map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [1]</td> </tr> <tr> <td>...</td> <td>...</td> </tr> <tr> <td>11111b</td> <td>map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [31]</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	00000b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [0]	00001b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [1]	...	...	11111b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [31]
<u>Bits</u>	<u>Definition</u>										
00000b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [0]										
00001b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [1]										
...	...										
11111b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [31]										

### SMIx4C SciMap3

Bits	Description										
31:29	Reserved.										
28:24	<b>SciMap15.</b> Read-write. Reset: 0. Specifies the mapping of GEVENT15 to one of the 32 bits in <a href="#">SMIx00 [Event_Status]</a> . See: SciMap12.										
23:21	Reserved.										
20:16	<b>SciMap14.</b> Read-write. Reset: 0. Specifies the mapping of GEVENT14 to one of the 32 bits in <a href="#">SMIx00 [Event_Status]</a> . See: SciMap12.										
15:13	Reserved.										
12:8	<b>SciMap13.</b> Read-write. Reset: 0. Specifies the mapping of GEVENT13 to one of the 32 bits in <a href="#">SMIx00 [Event_Status]</a> . See: SciMap12.										
7:5	Reserved.										
4:0	<p><b>SciMap12.</b> Read-write. Reset: 0. Specifies the mapping of GEVENT12 to one of the 32 bits in <a href="#">SMIx00 [Event_Status]</a>.</p> <table border="0"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>00000b</td> <td>map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [0]</td> </tr> <tr> <td>00001b</td> <td>map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [1]</td> </tr> <tr> <td>...</td> <td>...</td> </tr> <tr> <td>11111b</td> <td>map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [31]</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	00000b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [0]	00001b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [1]	...	...	11111b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [31]
<u>Bits</u>	<u>Definition</u>										
00000b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [0]										
00001b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [1]										
...	...										
11111b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [31]										

### SMIx50 SciMap4

Bits	Description
31:29	Reserved.
28:24	<b>SciMap19.</b> Read-write. Reset: 0. Specifies the mapping of GEVENT19 to one of the 32 bits in <a href="#">SMIx00 [Event_Status]</a> . See: SciMap16.
23:21	Reserved.
20:16	<b>SciMap18.</b> Read-write. Reset: 0. Specifies the mapping of GEVENT18 to one of the 32 bits in <a href="#">SMIx00 [Event_Status]</a> . See: SciMap16.
15:13	Reserved.
12:8	<b>SciMap17.</b> Read-write. Reset: 0. Specifies the mapping of GEVENT17 to one of the 32 bits in <a href="#">SMIx00 [Event_Status]</a> . See: SciMap16.

7:5	Reserved.										
4:0	<p><b>SciMap16.</b> Read-write. Reset: 0. Specifies the mapping of GEVENT16 to one of the 32 bits in <a href="#">SMIx00 [Event_Status]</a>.</p> <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>00000b</td> <td>map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [0]</td> </tr> <tr> <td>00001b</td> <td>map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [1]</td> </tr> <tr> <td>...</td> <td>...</td> </tr> <tr> <td>11111b</td> <td>map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [31]</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	00000b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [0]	00001b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [1]	...	...	11111b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [31]
<u>Bits</u>	<u>Definition</u>										
00000b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [0]										
00001b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [1]										
...	...										
11111b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [31]										

### SMIx54 SciMap5

Bits	Description										
31:29	Reserved.										
28:24	<b>SciMap23.</b> Read-write. Reset: 0. Specifies the mapping of GEVENT23 to one of the 32 bits in <a href="#">SMIx00 [Event_Status]</a> . See: SciMap20.										
23:21	Reserved.										
20:16	<b>SciMap22.</b> Read-write. Reset: 0. Specifies the mapping of GEVENT22 to one of the 32 bits in <a href="#">SMIx00 [Event_Status]</a> . See: SciMap20.										
15:13	Reserved.										
12:8	<b>SciMap21.</b> Read-write. Reset: 0. Specifies the mapping of GEVENT21 to one of the 32 bits in <a href="#">SMIx00 [Event_Status]</a> . See: SciMap20.										
7:5	Reserved.										
4:0	<p><b>SciMap20.</b> Read-write. Reset: 0. Specifies the mapping of GEVENT20 to one of the 32 bits in <a href="#">SMIx00 [Event_Status]</a>.</p> <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>00000b</td> <td>map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [0]</td> </tr> <tr> <td>00001b</td> <td>map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [1]</td> </tr> <tr> <td>...</td> <td>...</td> </tr> <tr> <td>11111b</td> <td>map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [31]</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	00000b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [0]	00001b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [1]	...	...	11111b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [31]
<u>Bits</u>	<u>Definition</u>										
00000b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [0]										
00001b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [1]										
...	...										
11111b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [31]										

### SMIx58 SciMap6

Bits	Description
31:21	Reserved.
20:16	<b>SciMap26.</b> Read-write. Reset: 0. Specifies the mapping of USB_PME (device 16h) to one of the 32 bits in <a href="#">SMIx00 [Event_Status]</a> . See: SciMap24.
15:13	Reserved.
12:8	<b>SciMap25.</b> Read-write. Reset: 0. Specifies the mapping of USB_PME (device 13h) to one of the 32 bits in <a href="#">SMIx00 [Event_Status]</a> . See: SciMap24.

7:5	Reserved.										
4:0	<b>SciMap24.</b> Read-write. Reset: 0. Specifies the mapping of USB_PME (device 12h) to one of the 32 bits in <a href="#">SMIx00 [Event_Status]</a> . <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>00000b</td> <td>map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [0]</td> </tr> <tr> <td>00001b</td> <td>map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [1]</td> </tr> <tr> <td>...</td> <td>...</td> </tr> <tr> <td>11111b</td> <td>map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [31]</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	00000b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [0]	00001b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [1]	...	...	11111b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [31]
<u>Bits</u>	<u>Definition</u>										
00000b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [0]										
00001b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [1]										
...	...										
11111b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [31]										

**SMIx5C SciMap7**

Bits	Description
31:0	Reserved.

**SMIx60 SciMap8**

Bits	Description
31:0	Reserved.

**SMIx64 SciMap9**

Bits	Description										
31:21	Reserved.										
20:16	<b>SciMap38.</b> Read-write. Reset: 0. Specifies the mapping of SATA1_PME to one of the 32 bits in <a href="#">SMIx00 [Event_Status]</a> . See: <a href="#">SciMap36</a> .										
15:13	Reserved.										
12:8	<b>SciMap37.</b> Read-write. Reset: 0. Specifies the mapping of SATA0_PME to one of the 32 bits in <a href="#">SMIx00 [Event_Status]</a> . See: <a href="#">SciMap36</a> .										
7:5	Reserved.										
4:0	<b>SciMap36.</b> Read-write. Reset: 0. Specifies the mapping of HD_Audio_PME to one of the 32 bits in <a href="#">SMIx00 [Event_Status]</a> . <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>00000b</td> <td>map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [0]</td> </tr> <tr> <td>00001b</td> <td>map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [1]</td> </tr> <tr> <td>...</td> <td>...</td> </tr> <tr> <td>11111b</td> <td>map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [31]</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	00000b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [0]	00001b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [1]	...	...	11111b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [31]
<u>Bits</u>	<u>Definition</u>										
00000b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [0]										
00001b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [1]										
...	...										
11111b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [31]										

**SMIx68 SciMap10**

Bits	Description
31:29	Reserved.
28:24	<b>SciMap43.</b> Read-write. Reset: 0. Specifies the mapping of WAKE# pin to one of the 32 bits in <a href="#">SMIx00 [Event_Status]</a> . See: <a href="#">SciMap40</a> .

23:21	Reserved.										
20:16	<b>SciMap42</b> . Read-write. Reset: 0. Specifies the mapping of CIR_PME to one of the 32 bits in <a href="#">SMIx00 [Event_Status]</a> . See: SciMap40.										
15:13	Reserved.										
12:8	<b>SciMap41</b> . Read-write. Reset: 0. Specifies the mapping of EC1_PME to one of the 32 bits in <a href="#">SMIx00 [Event_Status]</a> . See: SciMap40.										
7:5	Reserved.										
4:0	<b>SciMap40</b> . Read-write. Reset: 0. Specifies the mapping of EC0_PME to one of the 32 bits in <a href="#">SMIx00 [Event_Status]</a> . <table border="0" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; border-bottom: 1px solid black;">Bits</th> <th style="text-align: left; border-bottom: 1px solid black;">Definition</th> </tr> </thead> <tbody> <tr> <td>00000b</td> <td>map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [0]</td> </tr> <tr> <td>00001b</td> <td>map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [1]</td> </tr> <tr> <td>...</td> <td>...</td> </tr> <tr> <td>11111b</td> <td>map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [31]</td> </tr> </tbody> </table>	Bits	Definition	00000b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [0]	00001b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [1]	...	...	11111b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [31]
Bits	Definition										
00000b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [0]										
00001b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [1]										
...	...										
11111b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [31]										

### SMIx6C SciMap11

---

Bits	Description										
31:29	Reserved.										
28:24	<b>SciMap47</b> . Read-write. Reset: 0. Specifies the mapping of SMBUS0 Interrupt event to one of the 32 bits in <a href="#">SMIx00 [Event_Status]</a> . See: SciMap44.										
23:21	Reserved.										
20:16	<b>SciMap46</b> . Read-write. Reset: 0. Specifies the mapping of ASF slave interrupt event to one of the 32 bits in <a href="#">SMIx00 [Event_Status]</a> . See: SciMap44.										
15:13	Reserved.										
12:8	<b>SciMap45</b> . Read-write. Reset: 0. Specifies the mapping of ASF master interrupt event to one of the 32 bits in <a href="#">SMIx00 [Event_Status]</a> . See: SciMap44.										
7:5	Reserved.										
4:0	<b>SciMap44</b> . Read-write. Reset: 0. Specifies the mapping of internal FAN/THERMAL event to one of the 32 bits in <a href="#">SMIx00 [Event_Status]</a> . <table border="0" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; border-bottom: 1px solid black;">Bits</th> <th style="text-align: left; border-bottom: 1px solid black;">Definition</th> </tr> </thead> <tbody> <tr> <td>00000b</td> <td>map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [0]</td> </tr> <tr> <td>00001b</td> <td>map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [1]</td> </tr> <tr> <td>...</td> <td>...</td> </tr> <tr> <td>11111b</td> <td>map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [31]</td> </tr> </tbody> </table>	Bits	Definition	00000b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [0]	00001b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [1]	...	...	11111b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [31]
Bits	Definition										
00000b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [0]										
00001b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [1]										
...	...										
11111b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [31]										

### SMIx70 SciMap12

---

Bits	Description
31:29	Reserved.
28:24	<b>SciMap51</b> . Read-write. Reset: 0. Specifies the mapping of PWRBTN status to one of the 32 bits in <a href="#">SMIx00 [Event_Status]</a> . See: SciMap48.
23:21	Reserved.

20:16	<b>SciMap50.</b> Read-write. Reset: 0. Specifies the mapping of LLB# to one of the 32 bits in <a href="#">SMIx00 [Event_Status]</a> . See: SciMap48.										
15:13	Reserved.										
12:8	<b>SciMap49.</b> Read-write. Reset: 0. Specifies the mapping of internal traffic monitor to one of the 32 bits in <a href="#">SMIx00 [Event_Status]</a> . See: SciMap48.										
7:5	Reserved.										
4:0	<p><b>SciMap48.</b> Read-write. Reset: 0. Specifies the mapping of TWARDN pin to one of the 32 bits in <a href="#">SMIx00 [Event_Status]</a>.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00000b</td> <td>map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [0]</td> </tr> <tr> <td>00001b</td> <td>map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [1]</td> </tr> <tr> <td>...</td> <td>...</td> </tr> <tr> <td>11111b</td> <td>map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [31]</td> </tr> </tbody> </table>	Bits	Definition	00000b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [0]	00001b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [1]	...	...	11111b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [31]
Bits	Definition										
00000b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [0]										
00001b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [1]										
...	...										
11111b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [31]										

**SMIx74 SciMap13**

Bits	Description										
31:29	Reserved.										
28:24	<b>SciMap55.</b> Read-write. Reset: 0. Specifies the mapping of RAS_event status to one of the 32 bits in <a href="#">SMIx00 [Event_Status]</a> . See: SciMap52.										
23:21	Reserved.										
20:16	<b>SciMap54.</b> Read-write. Reset: 0. Specifies the mapping of SCI assertion message from APU to one of the 32 bits in <a href="#">SMIx00 [Event_Status]</a> . See: SciMap52.										
15:13	Reserved.										
12:8	<b>SciMap53.</b> Read-write. Reset: 0. Specifies the mapping of HW assertion message from APU to one of the 32 bits in <a href="#">SMIx00 [Event_Status]</a> . See: SciMap52.										
7:5	Reserved.										
4:0	<p><b>SciMap52.</b> Read-write. Reset: 0. Specifies the mapping of PROHOT# pin to one of the 32 bits in <a href="#">SMIx00 [Event_Status]</a>.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00000b</td> <td>map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [0]</td> </tr> <tr> <td>00001b</td> <td>map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [1]</td> </tr> <tr> <td>...</td> <td>...</td> </tr> <tr> <td>11111b</td> <td>map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [31]</td> </tr> </tbody> </table>	Bits	Definition	00000b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [0]	00001b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [1]	...	...	11111b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [31]
Bits	Definition										
00000b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [0]										
00001b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [1]										
...	...										
11111b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [31]										

**SMIx78 SciMap14**

Bits	Description
31:29	Reserved.
28:24	Reserved.
23:21	Reserved.
20:16	<b>SciMap58.</b> Read-write. Reset: 0. Specifies the mapping of AcDcTimerEvent to one of the 32 bits in <a href="#">SMIx00 [Event_Status]</a> . See: SciMap56.

15:5	Reserved.										
4:0	<p><b>SciMap56.</b> Read-write. Reset: 0. Specifies the mapping of XHC0 (USB3 controller 0) pin to one of the 32 bits in <a href="#">SMIx00 [Event_Status]</a>.</p> <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>00000b</td> <td>map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [0]</td> </tr> <tr> <td>00001b</td> <td>map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [1]</td> </tr> <tr> <td>...</td> <td>...</td> </tr> <tr> <td>11111b</td> <td>map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [31]</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	00000b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [0]	00001b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [1]	...	...	11111b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [31]
<u>Bits</u>	<u>Definition</u>										
00000b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [0]										
00001b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [1]										
...	...										
11111b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [31]										

### SMIx7C SciMap15

Bits	Description										
31:29	Reserved.										
28:24	<p><b>SciMap63.</b> Read-write. Reset: 0. Specifies the mapping of TempStatus5 (auto-polling temperature) to one of the 32 bits in <a href="#">SMIx00 [Event_Status]</a>.</p> <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>00000b</td> <td>map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [0]</td> </tr> <tr> <td>00001b</td> <td>map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [1]</td> </tr> <tr> <td>...</td> <td>...</td> </tr> <tr> <td>11111b</td> <td>map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [31]</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	00000b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [0]	00001b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [1]	...	...	11111b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [31]
<u>Bits</u>	<u>Definition</u>										
00000b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [0]										
00001b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [1]										
...	...										
11111b	map to the input of <a href="#">SMIx00 [Event_Status]</a> bit [31]										
23:0	Reserved.										

### SMIx80 SmiStatus0

Bits	Description
31:27	Reserved.
26	<b>UsbWakup2Event26.</b> Read; write-1-to-clear. Reset: 0. Status of USB device 16h PME.
25	<b>UsbWakup1Event25.</b> Read; write-1-to-clear. Reset: 0. Status of USB device 13h PME.
24	<b>UsbWakup0Event24.</b> Read; write-1-to-clear. Reset: 0. Status of USB device 12h PME.
23	<b>Gevent23StatusEvent23.</b> Read; write-1-to-clear. Reset: 0. Status of Gevent23.
22	<b>Gevent22StatusEvent22.</b> Read; write-1-to-clear. Reset: 0. Status of Gevent22.
21	<b>Gevent21StatusEvent21.</b> Read; write-1-to-clear. Reset: 0. Status of Gevent21.
20	<b>Gevent20StatusEvent20.</b> Read; write-1-to-clear. Reset: 0. Status of Gevent20.
19	<b>Gevent19StatusEvent19.</b> Read; write-1-to-clear. Reset: 0. Status of Gevent19.
18	<b>Gevent18StatusEvent18.</b> Read; write-1-to-clear. Reset: 0. Status of Gevent18.
17	<b>Gevent17StatusEvent17.</b> Read; write-1-to-clear. Reset: 0. Status of Gevent17.
16	<b>Gevent16StatusEvent16.</b> Read; write-1-to-clear. Reset: 0. Status of Gevent16.
15	<b>Gevent15StatusEvent15.</b> Read; write-1-to-clear. Reset: 0. Status of Gevent15.
14	<b>Gevent14StatusEvent14.</b> Read; write-1-to-clear. Reset: 0. Status of Gevent14.
13	<b>Gevent13StatusEvent13.</b> Read; write-1-to-clear. Reset: 0. Status of Gevent13.
12	<b>Gevent12StatusEvent12.</b> Read; write-1-to-clear. Reset: 0. Status of Gevent12.
11	<b>Gevent11StatusEvent11.</b> Read; write-1-to-clear. Reset: 0. Status of Gevent11.

10	<b>Gevent10StatusEvent10.</b> Read; write-1-to-clear. Reset: 0. Status of Gevent10.
9	<b>Gevent9StatusEvent9.</b> Read; write-1-to-clear. Reset: 0. Status of Gevent9.
8	<b>Gevent8StatusEvent8.</b> Read; write-1-to-clear. Reset: 0. Status of Gevent8.
7	<b>Gevent7StatusEvent7.</b> Read; write-1-to-clear. Reset: 0. Status of Gevent7.
6	<b>Gevent6StatusEvent6.</b> Read; write-1-to-clear. Reset: 0. Status of Gevent6.
5	<b>Gevent5StatusEvent5.</b> Read; write-1-to-clear. Reset: 0. Status of Gevent5.
4	<b>Gevent4StatusEvent4.</b> Read; write-1-to-clear. Reset: 0. Status of Gevent4.
3	<b>Gevent3StatusEvent3.</b> Read; write-1-to-clear. Reset: 0. Status of Gevent3.
2	<b>Gevent2StatusEvent2.</b> Read; write-1-to-clear. Reset: 0. Status of Gevent2.
1	<b>Gevent1StatusEvent1.</b> Read; write-1-to-clear. Reset: 0. Status of Gevent1.
0	<b>Gevent0StatusEvent0.</b> Read; write-1-to-clear. Reset: 0. Status of Gevent0.

**SMIx84 SmiStatus1**

Bits	Description
31	<b>TempTsiStsEvent63.</b> Read; write-1-to-clear. Reset: 0. Status of TempTsi event.
30:27	Reserved.
26	<b>AcDcTimerEventEvent58.</b> Read; write-1-to-clear. Reset: 0. Status of AcDcTimer wake up event.
25	Reserved.
24	<b>Xhc0PmeEvent56.</b> Read; write-1-to-clear. Reset: 0. Status of XHC0 (device 10h, function 0) PME.
23	<b>RasEvent55.</b> Read; write-1-to-clear. Reset: 0. BIOS: See <a href="#">2.15.3.9</a> . Internal devices SERR error status.
22	<b>ApuSciAssertionEvent54.</b> Read; write-1-to-clear. Reset: 0. Status of NB SCI request.
21	<b>ApuHwAssertionEvent53.</b> Read; write-1-to-clear. Reset: 0. Status of APU HW assertion.
20	<b>ProcHotEvent52.</b> Read; write-1-to-clear. Reset: 0. Status of ProcHot event.
19	<b>PwrButtonEvent51.</b> Read; write-1-to-clear. Reset: 0. Status of PwrButton (rising edge).
18	<b>IllbEvent50.</b> Read; write-1-to-clear. Reset: 0. Status of iLLB# assertion.
17	<b>TrafficMonitorIntrEvent49.</b> Read; write-1-to-clear. Reset: 0. Status of FCH Traffic Monitor Interrupt.
16	<b>TwarnEvent48.</b> Read; write-1-to-clear. Reset: 0. Status of FCH TWARN.
15	<b>Smbus0Event47.</b> Read; write-1-to-clear. Reset: 0. Status of FCH SMBUS0 master interrupt.
14	<b>AsfSlaveIntrEvent46.</b> Read; write-1-to-clear. Reset: 0. Status of FCH ASF slave interrupt.
13	<b>AsfMasterIntrEvent45.</b> Read; write-1-to-clear. Reset: 0. Status of FCH ASF master interrupt.
12	<b>FanThermalGeventEvent44.</b> Read; write-1-to-clear. Reset: 0. Status of FCH fan thermal.
11	<b>AltMmTimerStsEvent43.</b> Read; write-1-to-clear. Reset: 0. Status of AltHpetTimer Alarm.
10	<b>CirPmeEvent42.</b> Read; write-1-to-clear. Reset: 0. Status of FCH CIR PME.
9	<b>EcGevent1Event41.</b> Read; write-1-to-clear. Reset: 0. Status of FCH IMC Gevent1.
8	<b>EcGevent0Event40.</b> Read; write-1-to-clear. Reset: 0. Status of FCH IMC Gevent0.
7	Reserved.



6	<b>SataGevent1Event38.</b> Read; write-1-to-clear. Reset: 0. Status of FCH SATA Gevent1.
5	<b>SataGevent0Event37.</b> Read; write-1-to-clear. Reset: 0. Status of FCH SATA Gevent0.
4	<b>AzaliaPmeEvent36.</b> Read; write-1-to-clear. Reset: 0. Status of FCH HD Audio PME.
3:0	Reserved.

**SMI<sub>x</sub>88 SMIStatus2**

Bits	Description
31	<b>CmosEraseStsEvent95.</b> Read; write-1-to-clear. Reset: 0. Status of Cmos Erase event.
30	Reserved.
29	<b>AzFlrEvent93.</b> Read; write-1-to-clear. Reset: 0. Status of Azalia FLR event.
28	<b>SataFlrEvent92.</b> Read; write-1-to-clear. Reset: 0. Status of SATA FLR event.
27	<b>UsbFlrEvent91.</b> Read; write-1-to-clear. Reset: 0. Status of USB FLR event.
26	<b>Emulate64Event90.</b> Read; write-1-to-clear. Reset: 0. Status of Emulation IO Port 60h/64h.
25	<b>ThermaltripEvent89.</b> Read; write-1-to-clear. Reset: 0. Status of ThermalTrip event.
24:22	Reserved.
21	Reserved.
20	<b>PciSerrEvent84.</b> Read; write-1-to-clear. Reset: 0. Status of Serr assertion on PCI bus.
19	<b>ProtHotEvent83.</b> Read; write-1-to-clear. Reset: 0. Status of ProtHot event.
18	<b>VbatLowEvent82.</b> Read; write-1-to-clear. Reset: 0. Status of VBAT low.
17	Reserved.
16	<b>XhcErrEvent80.</b> Read; write-1-to-clear. Reset: 0. Status of XHC error.
15	<b>EcSmi0Event79.</b> Read; write-1-to-clear. Reset: 0. Status of IMC SMI request.
14	<b>Smbus0IntrEvent78.</b> Read; write-1-to-clear. Reset: 0. Status of SMBUS0 interrupt request.
13	<b>SerialIrqSmiEvent77.</b> Read; write-1-to-clear. Reset: 0. Status of SMI request from Serial IRQ.
12	<b>UsbSmiEvent76.</b> Read; write-1-to-clear. Reset: 0. Status of USB SMI request.
11	<b>SmiCmdportEvent75.</b> Read; write-1-to-clear. Reset: 0. Status of Writing SMI Command Port.
10	<b>PwrbtnEvent74.</b> Read; write-1-to-clear. Reset: 0. Status of Power Button being pressed.
9	<b>BiosRlsEvent73.</b> Read; write-1-to-clear. Reset: 0. Status of BIOS_RLS. See <a href="#">PMx74[BiosRls]</a> .
8	<b>GblRlsEvent72.</b> Read; write-1-to-clear. Reset: 0. Status of GBL event. See <a href="#">AcpiPm1CntBlkx00[GblRls]</a> .
7	<b>AcpiTimerEvent71.</b> Read; write-1-to-clear. Reset: 0. Status of ACPI timer rollover interrupt.
6	<b>RtcIrqEvent70.</b> Read; write-1-to-clear. Reset: 0. Status of RTC IRQ.
5	<b>ApuGpphpEvent69.</b> Read; write-1-to-clear. Reset: 0. Status of APU HP message.
4	<b>ApuGpppmeEvent68.</b> Read; write-1-to-clear. Reset: 0. Status of APU GPP PME message.
3	<b>SataAhciSmiEvent67.</b> Read; write-1-to-clear. Reset: 0. Status of SATA AHCI SMI request.
2	Reserved.
1	<b>SlpTypeEvent65.</b> Read; write-1-to-clear. Reset: 0. Status of writing <a href="#">AcpiPm1CntBlkx00[SlpTyp]</a> .
0	<b>KbRstEvent64.</b> Read; write-1-to-clear. Reset: 0. Status of KeyBoard Reset event.

**SMIx8C SMIStatus3**

Bits	Description
31:24	Reserved.
23	<b>Irq23TrappingEvent119.</b> Read; write-1-to-clear. Reset: 0. Status of IRQ23 request.
22	<b>Irq22TrappingEvent118.</b> Read; write-1-to-clear. Reset: 0. Status of IRQ22 request.
21	<b>Irq21TrappingEvent117.</b> Read; write-1-to-clear. Reset: 0. Status of IRQ21 request.
20	<b>Irq20TrappingEvent116.</b> Read; write-1-to-clear. Reset: 0. Status of IRQ20 request.
19	<b>Irq19TrappingEvent115.</b> Read; write-1-to-clear. Reset: 0. Status of IRQ19 request.
18	<b>Irq18TrappingEvent114.</b> Read; write-1-to-clear. Reset: 0. Status of IRQ18 request.
17	<b>Irq17TrappingEvent113.</b> Read; write-1-to-clear. Reset: 0. Status of IRQ17 request.
16	<b>Irq16TrappingEvent112.</b> Read; write-1-to-clear. Reset: 0. Status of IRQ16 request.
15	<b>Irq15TrappingEvent111.</b> Read; write-1-to-clear. Reset: 0. Status of IRQ15 request.
14	<b>Irq14TrappingEvent110.</b> Read; write-1-to-clear. Reset: 0. Status of IRQ14 request.
13	<b>Irq13TrappingEvent109.</b> Read; write-1-to-clear. Reset: 0. Status of IRQ13 request.
12	<b>Irq12TrappingEvent108.</b> Read; write-1-to-clear. Reset: 0. Status of IRQ12 request.
11	<b>Irq11TrappingEvent107.</b> Read; write-1-to-clear. Reset: 0. Status of IRQ11 request.
10	<b>Irq10TrappingEvent106.</b> Read; write-1-to-clear. Reset: 0. Status of IRQ10 request.
9	<b>Irq9TrappingEvent105.</b> Read; write-1-to-clear. Reset: 0. Status of IRQ9 request.
8	<b>Irq8TrappingEvent104.</b> Read; write-1-to-clear. Reset: 0. Status of IRQ8 request.
7	<b>Irq7TrappingEvent103.</b> Read; write-1-to-clear. Reset: 0. Status of IRQ7 request.
6	<b>Irq6TrappingEvent102.</b> Read; write-1-to-clear. Reset: 0. Status of IRQ6 request.
5	<b>Irq5TrappingEvent101.</b> Read; write-1-to-clear. Reset: 0. Status of IRQ5 request.
4	<b>Irq4TrappingEvent100.</b> Read; write-1-to-clear. Reset: 0. Status of IRQ4 request.
3	<b>Irq3TrappingEvent99.</b> Read; write-1-to-clear. Reset: 0. Status of IRQ3 request.
2	<b>Irq2TrappingEvent98.</b> Read; write-1-to-clear. Reset: 0. Status of IRQ2 request.
1	<b>Irq1TrappingEvent97.</b> Read; write-1-to-clear. Reset: 0. Status of IRQ1 request.
0	<b>Irq0TrappingEvent96.</b> Read; write-1-to-clear. Reset: 0. Status of IRQ0 request.

**SMIx90 SMIStatus4**

Bits	Description
31	<b>CfgTrapping3Event159.</b> Read; write-1-to-clear. Reset: 0. Status of PCI configuration cycle Trapping3 SMI request.
30	<b>CfgTrapping2Event158.</b> Read; write-1-to-clear. Reset: 0. Status of PCI configuration cycle Trapping2 SMI request.
29	<b>CfgTrapping1Event157.</b> Read; write-1-to-clear. Reset: 0. Status of PCI configuration cycle Trapping1 SMI request.
28	<b>CfgTrapping0Event156.</b> Read; write-1-to-clear. Reset: 0. Status of PCI configuration cycle Trapping0 SMI request.

27	<b>MemTrapping3Event155.</b> Read; write-1-to-clear. Reset: 0. Status of memory Trapping3 SMI request.
26	<b>MemTrapping2Event154.</b> Read; write-1-to-clear. Reset: 0. Status of memory Trapping2 SMI request.
25	<b>MemTrapping1Event153.</b> Read; write-1-to-clear. Reset: 0. Status of memory Trapping1 SMI request.
24	<b>MemTrapping0Event152.</b> Read; write-1-to-clear. Reset: 0. Status of memory Trapping0 SMI request.
23	<b>IoTrapping3Event151.</b> Read; write-1-to-clear. Reset: 0. Status of IO Trapping3 SMI request.
22	<b>IoTrapping2Event150.</b> Read; write-1-to-clear. Reset: 0. Status of IO Trapping2 SMI request.
21	<b>IoTrapping1Event149.</b> Read; write-1-to-clear. Reset: 0. Status of IO Trapping1 SMI request.
20	<b>IoTrapping0Event148.</b> Read; write-1-to-clear. Reset: 0. Status of IO Trapping0 SMI request.
19	<b>PstateChangeEvent147.</b> Read; write-1-to-clear. Reset: 0. Status of P state request.
18	<b>PstateChangeEvent146.</b> Read; write-1-to-clear. Reset: 0. Status of P state request.
17	<b>SoftResetEvent145.</b> Read; write-1-to-clear. Reset: 0. Status of soft reset request. The soft request can be: <ul style="list-style-type: none"> <li>• CF9h PCI reset.</li> <li>• ASF remote reset.</li> <li>• sync flood reset.</li> <li>• KBRST.</li> <li>• watchdog timer reset.</li> </ul>
16	<b>AbSmiTrapEvent144.</b> Read; write-1-to-clear. Reset: 0. Status of AB SMI trapping request.
15	<b>LongTimerEvent143.</b> Read; write-1-to-clear. Reset: 0. Status of Long Timer SMI request.
14	<b>ShortTimerEvent142.</b> Read; write-1-to-clear. Reset: 0. Status of Short Timer SMI request.
13	<b>CStateMsgEvent141.</b> Read; write-1-to-clear. Reset: 0. Status of C State Change message request.
12	<b>Fake2StsEvent140.</b> Read; write-1-to-clear. Reset: 0. Status of Fake2.
11	<b>Fake1StsEvent139.</b> Read; write-1-to-clear. Reset: 0. Status of Fake1.
10	<b>Fake0StsEvent138.</b> Read; write-1-to-clear. Reset: 0. Status of Fake0.
9:6	Reserved.
5	<b>Fanin0stsEvent133.</b> Read; write-1-to-clear. Reset: 0. Status of FanIn0 event.
4:0	Reserved.

### SMIx94 SmiPointer

This register is meant as a faster mechanism to locate the SMI source. BIOS can examine this register to find out the SMI source instead of reading [SMIx80 \[SmiStatus0\]](#) through [SMIx90 \[SmiStatus4\]](#) individually.

Bits	Description
15:6	Reserved.
5	<b>SmiStatusSource4.</b> Read-only. Reset: 0. Indicates whether the SMI source is from <a href="#">SMIx90 [SmiStatus4]</a> if the corresponding SMI enable is selected.
4	<b>SmiStatusSource3.</b> Read-only. Reset: 0. Indicates whether the SMI source is from <a href="#">SMIx8C [SmiStatus3]</a> if the corresponding SMI enable is selected.

3	<b>SmiStatusSource2</b> . Read-only. Reset: 0. Indicates whether the SMI source is from <a href="#">SMIx88 [SmiStatus2]</a> if the corresponding SMI enable is selected.
2	<b>SmiStatusSource1</b> . Read-only. Reset: 0. Indicates whether the SMI source is from <a href="#">SMIx84 [SmiStatus1]</a> if the corresponding SMI enable is selected.
1	<b>SmiStatusSource0</b> . Read-only. Reset: 0. Indicates whether the SMI source is from <a href="#">SMIx80 [SmiStatus0]</a> if the corresponding SMI enable is selected.
0	<b>SmiSciSource</b> . Read-only. Reset: 0. Indicates whether the SMI source is from <a href="#">SMIx10 [SmiSciStatus]</a> .

### SMIx96 SmiTimer

There are two SMI timers: the short timer runs at 1 us unit time and the long timer runs at 1 ms unit time. This register is actually made up of two sets of registers depending on the setting of [SMIx98\[SmiTimerSel\]](#). If [SMIx98\[SmiTimerSel\]==0](#), then [SMIx96](#) is for the short timer. If [SMIx98\[SmiTimerSel\]==1](#), then [SMIx96](#) is for the long timer. The default setting selects this register as “SmiShortTimer”; software needs to set the “SmiTimerSel=1” to select this register as “SmiLongTimer”.

Bits	Description
15	<b>TimerEn</b> . Read-write. Reset: 0. 1=Enable the SMI short Timer or long timer, which is selected by <a href="#">SMIx98[SmiTimerSel]</a> . 0=Disable.
14:0	<b>SmiTimerCount</b> . Read-write. Reset: 0. Actual timer duration = TimerTime + 1 unit.

### SMIx98 SmiTrig0

Bit [23:0] defines the trigger mode for 24 GEVENTS in [SMIx80\[23:0\]](#). Note these are different from [SMIx08 \[SciTrig\]](#).

Bits	Description
31	<b>SmiEnB</b> . Read-write. Reset: 1. 0=Enable SMI function. 1=Disable.
30	Reserved.
29	<b>SmiTimerSel</b> . Read-write. Reset: 0. 0=Select <a href="#">SMIx96 [SmiTimer]</a> to be SmiShortTimer register. 1=Select <a href="#">SMIx96 [SmiTimer]</a> to be SmiLongTimer register.
28	Reserved.
27	<b>FakeSts2</b> . Read-write. Reset: 1. Program the value to emulate an SMI input event.
26	<b>FakeSts1</b> . Read-write. Reset: 1. Program the value to emulate an SMI input event.
25	<b>FakeSts0</b> . Read-write. Reset: 1. Program the value to emulate an SMI input event.
24	<b>TrappingIrqOnPic</b> . Read-write. Reset: 1. 0=SMI is generated when trapping IRQ0-15 of IOAPIC. 1=SMI is generated when trapping IRQ0-15 of PIC.
23	<b>SmiTrig23</b> . Read-write. Reset: 1. 0=Active low. 1=Active high.
22	<b>SmiTrig22</b> . Read-write. Reset: 1. 0=Active low. 1=Active high.
21	<b>SmiTrig21</b> . Read-write. Reset: 1. 0=Active low. 1=Active high.
20	<b>SmiTrig20</b> . Read-write. Reset: 1. 0=Active low. 1=Active high.
19	<b>SmiTrig19</b> . Read-write. Reset: 1. 0=Active low. 1=Active high.
18	<b>SmiTrig18</b> . Read-write. Reset: 1. 0=Active low. 1=Active high.

17	<b>SmiTrig17</b> . Read-write. Reset: 1. 0=Active low. 1=Active high.
16	<b>SmiTrig16</b> . Read-write. Reset: 1. 0=Active low. 1=Active high.
15	<b>SmiTrig15</b> . Read-write. Reset: 1. 0=Active low. 1=Active high.
14	<b>SmiTrig14</b> . Read-write. Reset: 1. 0=Active low. 1=Active high.
13	<b>SmiTrig13</b> . Read-write. Reset: 1. 0=Active low. 1=Active high.
12	<b>SmiTrig12</b> . Read-write. Reset: 1. 0=Active low. 1=Active high.
11	<b>SmiTrig11</b> . Read-write. Reset: 1. 0=Active low. 1=Active high.
10	<b>SmiTrig10</b> . Read-write. Reset: 1. 0=Active low. 1=Active high.
9	<b>SmiTrig9</b> . Read-write. Reset: 1. 0=Active low. 1=Active high.
8	<b>SmiTrig8</b> . Read-write. Reset: 1. 0=Active low. 1=Active high.
7	<b>SmiTrig7</b> . Read-write. Reset: 1. 0=Active low. 1=Active high.
6	<b>SmiTrig6</b> . Read-write. Reset: 1. 0=Active low. 1=Active high.
5	<b>SmiTrig5</b> . Read-write. Reset: 1. 0=Active low. 1=Active high.
4	<b>SmiTrig4</b> . Read-write. Reset: 1. 0=Active low. 1=Active high.
3	<b>SmiTrig3</b> . Read-write. Reset: 1. 0=Active low. 1=Active high.
2	<b>SmiTrig2</b> . Read-write. Reset: 1. 0=Active low. 1=Active high.
1	<b>SmiTrig1</b> . Read-write. Reset: 1. 0=Active low. 1=Active high.
0	<b>SmiTrig0</b> . Read-write. Reset: 1. 0=Active low. 1=Active high.

### SMIx9C SmiTrig1

Bit [23:0] defines the trigger mode for 24 IRQs.

Bits	Description
31:0	Reserved.

### SMIxA0 SmiControl0

This register specifies the control mechanism for SMI sources in [SMIx80\[15:0\]](#).

Bits	Description
31:30	<b>Smicontrol15</b> . Read-write. Reset: 0. Control for GEVENT15. See: Smicontrol0.
29:28	<b>Smicontrol14</b> . Read-write. Reset: 0. Control for GEVENT14. See: Smicontrol0.
27:26	<b>Smicontrol13</b> . Read-write. Reset: 0. Control for GEVENT13. See: Smicontrol0.
25:24	<b>Smicontrol12</b> . Read-write. Reset: 0. Control for GEVENT12. See: Smicontrol0.
23:22	<b>Smicontrol11</b> . Read-write. Reset: 0. Control for GEVENT11. See: Smicontrol0.
21:20	<b>Smicontrol10</b> . Read-write. Reset: 0. Control for GEVENT10. See: Smicontrol0.
19:18	<b>Smicontrol9</b> . Read-write. Reset: 0. Control for GEVENT9. See: Smicontrol0.
17:16	<b>Smicontrol8</b> . Read-write. Reset: 0. Control for GEVENT8. See: Smicontrol0.
15:14	<b>Smicontrol7</b> . Read-write. Reset: 0. Control for GEVENT7. See: Smicontrol0.
13:12	<b>Smicontrol6</b> . Read-write. Reset: 0. Control for GEVENT6. See: Smicontrol0.
11:10	<b>Smicontrol5</b> . Read-write. Reset: 0. Control for GEVENT5. See: Smicontrol0.

9:8	<b>Smicontrol4</b> . Read-write. Reset: 0. Control for GEVENT4. See: Smicontrol0.										
7:6	<b>Smicontrol3</b> . Read-write. Reset: 0. Control for GEVENT3. See: Smicontrol0.										
5:4	<b>Smicontrol2</b> . Read-write. Reset: 0. Control for GEVENT2. See: Smicontrol0.										
3:2	<b>Smicontrol1</b> . Read-write. Reset: 0. Control for GEVENT1. See: Smicontrol0.										
1:0	<b>Smicontrol0</b> . Read-write. Reset: 0. Control for GEVENT0. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Disable</td> </tr> <tr> <td>01b</td> <td>SMI</td> </tr> <tr> <td>10b</td> <td>NMI</td> </tr> <tr> <td>11b</td> <td>IRQ13</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	00b	Disable	01b	SMI	10b	NMI	11b	IRQ13
<u>Bits</u>	<u>Definition</u>										
00b	Disable										
01b	SMI										
10b	NMI										
11b	IRQ13										

### SMIxA4 SmiControl1

This register specifies the control mechanism for SMI sources in [SMIx80](#)[31:16].

Bits	Description										
31:22	Reserved.										
21:20	<b>Smicontrol26</b> . Read-write. Reset: 0. Control for USB_PME (Device 14h). See: Smicontrol16.										
19:18	<b>Smicontrol25</b> . Read-write. Reset: 0. Control for USB_PME (Device 13h). See: Smicontrol16.										
17:16	<b>Smicontrol24</b> . Read-write. Reset: 0. Control for USB_PME (Device 12h). See: Smicontrol16.										
15:14	<b>Smicontrol23</b> . Read-write. Reset: 0. Control for GEVENT23. See: Smicontrol16.										
13:12	<b>Smicontrol22</b> . Read-write. Reset: 0. Control for GEVENT22. See: Smicontrol16.										
11:10	<b>Smicontrol21</b> . Read-write. Reset: 0. Control for GEVENT21. See: Smicontrol16.										
9:8	<b>Smicontrol20</b> . Read-write. Reset: 0. Control for GEVENT20. See: Smicontrol16.										
7:6	<b>Smicontrol19</b> . Read-write. Reset: 0. Control for GEVENT19. See: Smicontrol16.										
5:4	<b>Smicontrol18</b> . Read-write. Reset: 0. Control for GEVENT18. See: Smicontrol16.										
3:2	<b>Smicontrol17</b> . Read-write. Reset: 0. Control for GEVENT17. See: Smicontrol16.										
1:0	<b>Smicontrol16</b> . Read-write. Reset: 0. Control for GEVENT16. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Disable</td> </tr> <tr> <td>01b</td> <td>SMI</td> </tr> <tr> <td>10b</td> <td>NMI</td> </tr> <tr> <td>11b</td> <td>IRQ13</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	00b	Disable	01b	SMI	10b	NMI	11b	IRQ13
<u>Bits</u>	<u>Definition</u>										
00b	Disable										
01b	SMI										
10b	NMI										
11b	IRQ13										

### SMIxA8 SmiControl2

This register specifies the control mechanism for SMI sources in [SMIx84](#)[15:0].

Bits	Description
31:30	<b>Smicontrol47</b> . Read-write. Reset: 0. Control for SMBUS0 interrupt. See: Smicontrol36.
29:28	<b>Smicontrol46</b> . Read-write. Reset: 0. Control for ASF Slave interrupt. See: Smicontrol36.
27:26	<b>Smicontrol45</b> . Read-write. Reset: 0. Control for ASF Master interrupt. See: Smicontrol36.
25:24	<b>Smicontrol44</b> . Read-write. Reset: 0. Control for fan thermal GEvent. See: Smicontrol36.
23:22	Reserved.

21:20	<b>Smicontrol42</b> . Read-write. Reset: 0. Control for CIR PME. See: Smicontrol36.										
19:18	Reserved.										
17:16	<b>Smicontrol40</b> . Read-write. Reset: 0. Control for IMC Gevent0. See: Smicontrol36.										
15:14	Reserved.										
13:12	<b>Smicontrol38</b> . Read-write. Reset: 0. Control for SATA Gevent1. See: Smicontrol36.										
11:10	<b>Smicontrol37</b> . Read-write. Reset: 0. Control for SATA Gevent0. See: Smicontrol36.										
9:8	<b>Smicontrol36</b> . Read-write. Reset: 0. Control for PME from HD Audio. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Disable</td> </tr> <tr> <td>01b</td> <td>SMI</td> </tr> <tr> <td>10b</td> <td>NMI</td> </tr> <tr> <td>11b</td> <td>IRQ13</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	00b	Disable	01b	SMI	10b	NMI	11b	IRQ13
<u>Bits</u>	<u>Definition</u>										
00b	Disable										
01b	SMI										
10b	NMI										
11b	IRQ13										
7:4	Reserved.										
3:0	Reserved.										

### SMIxAC SmiControl3

This register specifies the control mechanism for SMI sources in [SMIx84](#)[31:16].

Bits	Description										
31:30	<b>Smicontrol63</b> . Read-write. Reset: 0. Control for TempTsi event. See: Smicontrol48.										
29:22	Reserved.										
21:20	<b>Smicontrol58</b> . Read-write. Reset: 0. Control for AcDcTimer wake up event (Wake Device in ACPI4.0). See: Smicontrol48.										
19:18	Reserved.										
17:16	<b>Smicontrol56</b> . Read-write. Reset: 0. Control for XHC0 (device 10h, function 0) PME. See: Smicontrol48.										
15:14	<b>Smicontrol55</b> . Read-write. Reset: 0. Control for internal devices SERR error status. See: Smicontrol48.										
13:12	<b>Smicontrol54</b> . Read-write. Reset: 0. Control for APU SCI request. See: Smicontrol48.										
11:10	<b>Smicontrol53</b> . Read-write. Reset: 0. Control for APU HW assertion. See: Smicontrol48.										
9:8	<b>Smicontrol52</b> . Read-write. Reset: 0. Control for ProcHot event. See: Smicontrol48.										
7:6	<b>Smicontrol51</b> . Read-write. Reset: 0. Control for Power button event. See: Smicontrol48.										
5:4	<b>Smicontrol50</b> . Read-write. Reset: 0. Control for iLLB#. See: Smicontrol48.										
3:2	<b>Smicontrol49</b> . Read-write. Reset: 0. Control for internal Traffic monitor interrupt. See: Smicontrol48.										
1:0	<b>Smicontrol48</b> . Read-write. Reset: 0. Control for TWARDN#. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Disable</td> </tr> <tr> <td>01b</td> <td>SMI</td> </tr> <tr> <td>10b</td> <td>NMI</td> </tr> <tr> <td>11b</td> <td>IRQ13</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	00b	Disable	01b	SMI	10b	NMI	11b	IRQ13
<u>Bits</u>	<u>Definition</u>										
00b	Disable										
01b	SMI										
10b	NMI										
11b	IRQ13										

**SMIxB0 SmiControl4**

This register specifies the control mechanism for SMI sources in [SMIxB0\[15:0\]](#).

Bits	Description										
31:30	<b>Smicontrol79</b> . Read-write. Reset: 0. Control for IMC SMI request0. See: Smicontrol65.										
29:28	<b>Smicontrol78</b> . Read-write. Reset: 0. Control for SMBUS0 interrupt. See: Smicontrol65.										
27:26	<b>Smicontrol77</b> . Read-write. Reset: 0. Control for SMI request form serial IRQ. See: Smicontrol65.										
25:24	<b>Smicontrol76</b> . Read-write. Reset: 0. Control for USB SMI request. See: Smicontrol65.										
23:22	<b>Smicontrol75</b> . Read-write. Reset: 0. Control for writing SMI command port at <a href="#">SmiCmdBlkx00</a> . See: Smicontrol65.										
21:20	<b>Smicontrol74</b> . Read-write. Reset: 0. Control for power button being pressed. See: Smicontrol65.										
19:18	<b>Smicontrol73</b> . Read-write. Reset: 0. Control for writing <a href="#">PMx74[BiosRIs]</a> . See: Smicontrol65.										
17:16	<b>Smicontrol72</b> . Read-write. Reset: 0. Control for writing <a href="#">AcpiPm1CntBlkx00[GblRIs]</a> . See: Smicontrol65.										
15:14	<b>Smicontrol71</b> . Read-write. Reset: 0. Control for PM timer rollover. See: Smicontrol65.										
13:12	<b>Smicontrol70</b> . Read-write. Reset: 0. Control for RTC IRQ. See: Smicontrol65.										
11:10	<b>Smicontrol69</b> . Read-write. Reset: 0. Control for APU GPP hot plug. See: Smicontrol65.										
9:8	<b>Smicontrol68</b> . Read-write. Reset: 0. Control for APU GPP PME. See: Smicontrol65.										
7:6	<b>Smicontrol67</b> . Read-write. Reset: 0. Control for SATA AHCI event. See: Smicontrol65.										
5:4	Reserved.										
3:2	<b>Smicontrol65</b> . Read-write. Reset: 0. Control for writing <a href="#">AcpiPm1CntBlkx00[SlpTyp]</a> to put the system in S state. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Disable</td> </tr> <tr> <td>01b</td> <td>SMI</td> </tr> <tr> <td>10b</td> <td>NMI</td> </tr> <tr> <td>11b</td> <td>IRQ13</td> </tr> </tbody> </table>	Bits	Definition	00b	Disable	01b	SMI	10b	NMI	11b	IRQ13
Bits	Definition										
00b	Disable										
01b	SMI										
10b	NMI										
11b	IRQ13										
1:0	Reserved.										

**SMIxB4 SmiControl5**

This register specifies the control mechanism for SMI sources in [SMIxB4\[31:16\]](#).

Bits	Description
31:30	<b>Smicontrol95</b> . Read-write. Reset: 0. Control for Cmos Erase. See: Smicontrol80.
29:28	Reserved.
27:26	<b>Smicontrol93</b> . Read-write. Reset: 0. Control for HD audio FLR. See: Smicontrol80.
25:24	<b>Smicontrol92</b> . Read-write. Reset: 0. Control for SATA FLR. See: Smicontrol80.
23:22	<b>Smicontrol91</b> . Read-write. Reset: 0. Control for USB FLR. See: Smicontrol80.
21:20	<b>Smicontrol90</b> . Read-write. Reset: 0. Control for Emulation64. See: Smicontrol80.
19:18	<b>Smicontrol89</b> . Read-write. Reset: 0. Control for ThermalTrip# assertion. See: Smicontrol80.
17:12	Reserved.



11:10	Reserved.										
9:8	<b>Smicontrol84.</b> Read-write. Reset: 0. Control for SERR#. See: Smicontrol80.										
7:6	<b>Smicontrol83.</b> Read-write. Reset: 0. Control for ProcHot. See: Smicontrol80.										
5:4	<b>Smicontrol82.</b> Read-write. Reset: 0. Control for VBAT low. See: Smicontrol80.										
3:2	Reserved.										
1:0	<b>Smicontrol80.</b> Read-write. Reset: 0. Control for IMC SMI request1. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Disable</td> </tr> <tr> <td>01b</td> <td>SMI</td> </tr> <tr> <td>10b</td> <td>NMI</td> </tr> <tr> <td>11b</td> <td>IRQ13</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	00b	Disable	01b	SMI	10b	NMI	11b	IRQ13
<u>Bits</u>	<u>Definition</u>										
00b	Disable										
01b	SMI										
10b	NMI										
11b	IRQ13										

### SMIxB8 SmiControl6

This register specifies the control mechanism for SMI sources in [SMIx8C\[15:0\]](#).

Bits	Description										
31:30	<b>Smicontrol111.</b> Read-write. Reset: 0. Control for IRQ15. See: Smicontrol96.										
29:28	<b>Smicontrol110.</b> Read-write. Reset: 0. Control for IRQ14. See: Smicontrol96.										
27:26	<b>Smicontrol109.</b> Read-write. Reset: 0. Control for IRQ13. See: Smicontrol96.										
25:24	<b>Smicontrol108.</b> Read-write. Reset: 0. Control for IRQ12. See: Smicontrol96.										
23:22	<b>Smicontrol107.</b> Read-write. Reset: 0. Control for IRQ11. See: Smicontrol96.										
21:20	<b>Smicontrol106.</b> Read-write. Reset: 0. Control for IRQ10. See: Smicontrol96.										
19:18	<b>Smicontrol105.</b> Read-write. Reset: 0. Control for IRQ9. See: Smicontrol96.										
17:16	<b>Smicontrol104.</b> Read-write. Reset: 0. Control for IRQ8. See: Smicontrol96.										
15:14	<b>Smicontrol103.</b> Read-write. Reset: 0. Control for IRQ7. See: Smicontrol96.										
13:12	<b>Smicontrol102.</b> Read-write. Reset: 0. Control for IRQ6. See: Smicontrol96.										
11:10	<b>Smicontrol101.</b> Read-write. Reset: 0. Control for IRQ5. See: Smicontrol96.										
9:8	<b>Smicontrol100.</b> Read-write. Reset: 0. Control for IRQ4. See: Smicontrol96.										
7:6	<b>Smicontrol99.</b> Read-write. Reset: 0. Control for IRQ3. See: Smicontrol96.										
5:4	<b>Smicontrol98.</b> Read-write. Reset: 0. Control for IRQ2. See: Smicontrol96.										
3:2	<b>Smicontrol97.</b> Read-write. Reset: 0. Control for IRQ1. See: Smicontrol96.										
1:0	<b>Smicontrol96.</b> Read-write. Reset: 0. Control for IRQ0. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Disable</td> </tr> <tr> <td>01b</td> <td>SMI</td> </tr> <tr> <td>10b</td> <td>NMI</td> </tr> <tr> <td>11b</td> <td>IRQ13</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	00b	Disable	01b	SMI	10b	NMI	11b	IRQ13
<u>Bits</u>	<u>Definition</u>										
00b	Disable										
01b	SMI										
10b	NMI										
11b	IRQ13										

### SMIxBC SmiControl7

This register specifies the control mechanism for SMI sources in [SMIx8C\[31:16\]](#).

Bits	Description										
31:16	Reserved.										
15:14	<b>Smicontrol119.</b> Read-write. Reset: 0. Control for IRQ23. See: Smicontrol112.										
13:12	<b>Smicontrol118.</b> Read-write. Reset: 0. Control for IRQ22. See: Smicontrol112.										
11:10	<b>Smicontrol117.</b> Read-write. Reset: 0. Control for IRQ21. See: Smicontrol112.										
9:8	<b>Smicontrol116.</b> Read-write. Reset: 0. Control for IRQ20. See: Smicontrol112.										
7:6	<b>Smicontrol115.</b> Read-write. Reset: 0. Control for IRQ19. See: Smicontrol112.										
5:4	<b>Smicontrol114.</b> Read-write. Reset: 0. Control for IRQ18. See: Smicontrol112.										
3:2	<b>Smicontrol113.</b> Read-write. Reset: 0. Control for IRQ17. See: Smicontrol112.										
1:0	<b>Smicontrol112.</b> Read-write. Reset: 0. Control for IRQ16. <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Disable</td> </tr> <tr> <td>01b</td> <td>SMI</td> </tr> <tr> <td>10b</td> <td>NMI</td> </tr> <tr> <td>11b</td> <td>IRQ 13</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	00b	Disable	01b	SMI	10b	NMI	11b	IRQ 13
<u>Bits</u>	<u>Definition</u>										
00b	Disable										
01b	SMI										
10b	NMI										
11b	IRQ 13										

### SMIxC0 SmiControl8

This register specifies the control mechanism for SMI sources in [SMIx90](#)[15:0].

Bits	Description										
31:30	<b>Smicontrol143.</b> Read-write. Reset: 0. Control for Long timer. See: Smicontrol133.										
29:28	<b>Smicontrol142.</b> Read-write. Reset: 0. Control for Short timer. See: Smicontrol133.										
27:26	<b>Smicontrol141.</b> Read-write. Reset: 0. Control for C state Message. See: Smicontrol133.										
25:24	<b>Smicontrol140.</b> Read-write. Reset: 0. Control for <a href="#">SMIx98</a> [FakeSts2] generated interrupts. See: Smicontrol133.										
23:22	<b>Smicontrol139.</b> Read-write. Reset: 0. Control for <a href="#">SMIx98</a> [FakeSts1] generated interrupts. See: Smicontrol133.										
21:20	<b>Smicontrol138.</b> Read-write. Reset: 0. Control for <a href="#">SMIx98</a> [FakeSts0] generated interrupts. See: Smicontrol133.										
19:12	Reserved.										
11:10	<b>Smicontrol133.</b> Read-write. Reset: 0. Control for Fan0 Tach too slow event. <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Disable</td> </tr> <tr> <td>01b</td> <td>SMI</td> </tr> <tr> <td>10b</td> <td>NMI</td> </tr> <tr> <td>11b</td> <td>IRQ 13</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	00b	Disable	01b	SMI	10b	NMI	11b	IRQ 13
<u>Bits</u>	<u>Definition</u>										
00b	Disable										
01b	SMI										
10b	NMI										
11b	IRQ 13										
9:0	Reserved.										

### SMIxC4 SmiControl9

This register specifies the control mechanism for SMI sources in [SMIx90](#)[31:16].

Bits	Description										
31:30	<b>Smicontrol159.</b> Read-write. Reset: 0. Control for configuration cycle trapping 3. See: Smicontrol144.										
29:28	<b>Smicontrol158.</b> Read-write. Reset: 0. Control for configuration cycle trapping 2. See: Smicontrol144.										
27:26	<b>Smicontrol157.</b> Read-write. Reset: 0. Control for configuration cycle trapping 1. See: Smicontrol144.										
25:24	<b>Smicontrol156.</b> Read-write. Reset: 0. Control for configuration cycle trapping 0. See: Smicontrol144.										
23:22	<b>Smicontrol155.</b> Read-write. Reset: 0. Control for memory trapping 3. See: Smicontrol144.										
21:20	<b>Smicontrol154.</b> Read-write. Reset: 0. Control for memory trapping 2. See: Smicontrol144.										
19:18	<b>Smicontrol153.</b> Read-write. Reset: 0. Control for memory trapping 1. See: Smicontrol144.										
17:16	<b>Smicontrol152.</b> Read-write. Reset: 0. Control for memory trapping 0. See: Smicontrol144.										
15:14	<b>Smicontrol151.</b> Read-write. Reset: 0. Control for IO trapping 3. See: Smicontrol144.										
13:12	<b>Smicontrol150.</b> Read-write. Reset: 0. Control for IO trapping 2. See: Smicontrol144.										
11:10	<b>Smicontrol149.</b> Read-write. Reset: 0. Control for IO trapping 1. See: Smicontrol144.										
9:8	<b>Smicontrol148.</b> Read-write. Reset: 0. Control for IO trapping 0. See: Smicontrol144.										
7:6	<b>Smicontrol147.</b> Read-write. Reset: 0. Control for P state message 2. See: Smicontrol144.										
5:4	<b>Smicontrol146.</b> Read-write. Reset: 0. Control for P state message 1. See: Smicontrol144.										
3:2	<b>Smicontrol145.</b> Read-write. Reset: 0. Control for P state message 0. See: Smicontrol144.										
1:0	<p><b>Smicontrol144.</b> Read-write. Reset: 0. Control for AB SMI trapping request.</p> <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Disable</td> </tr> <tr> <td>01b</td> <td>SMI</td> </tr> <tr> <td>10b</td> <td>NMI</td> </tr> <tr> <td>11b</td> <td>IRQ13</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	00b	Disable	01b	SMI	10b	NMI	11b	IRQ13
<u>Bits</u>	<u>Definition</u>										
00b	Disable										
01b	SMI										
10b	NMI										
11b	IRQ13										

### SMIxC8 IoTrapping0

Bits	Description
15:0	<b>IoTrapping0.</b> Read-write. Reset: 0. Specifies the IO address which causes SMI event.

### SMIxCA IoTrapping1

Bits	Description
15:0	<b>IoTrapping1.</b> Read-write. Reset: 0. Specifies the IO address which causes SMI event.

### SMIxCC IoTrapping2

Bits	Description
15:0	<b>IoTrapping2.</b> Read-write. Reset: 0. Specifies the IO address which causes SMI event.

**SMIxCE IoTrapping3**

Bits	Description
15:0	<b>IoTrapping3</b> . Read-write. Reset: 0. Specifies the IO address which causes SMI event.

**SMIxD0 MemTrapping0**

Bits	Description
31:2	<b>MemTrapping</b> . Read-write. Reset: 0. Specifies the 30-bit memory address which causes SMI event. The lowest 2 bits are ignored.
1	<b>MemRdOvrEn</b> . Read-write. Reset: 0. 1=Force read data to be replaced by <a href="#">SMIxD4 [MemRdOvrData0]</a> .
0	<b>MemTrappingRw</b> . Read-write. Reset: 0. 0=Trap on read access at the address specified by [Mem-Trapping]. 1=Trap on write access.

**SMIxD4 MemRdOvrData0**

Bits	Description
31:0	<b>MemRdOvrData</b> . Read-write. Reset: 0. This 32-bit data is used as the return data when the memory read trapping is enabled through <a href="#">SMIxD0[MemRdOvrEn]</a> .

**SMIxD8 MemTrapping1**

Bits	Description
31:2	<b>MemTrapping</b> . Read-write. Reset: 0. Specifies the 30-bit memory address which causes SMI event. The lowest 2 bits are ignored.
1	<b>MemRdOvrEn</b> . Read-write. Reset: 0. 1=Force read data to be replaced by <a href="#">SMIxDC [MemRdOvrData1]</a> .
0	<b>MemTrappingRw</b> . Read-write. Reset: 0. 0=Trap on read access at the address specified by [Mem-Trapping]. 1=Trap on write access.

**SMIxDC MemRdOvrData1**

Bits	Description
31:0	<b>MemRdOvrData</b> . Read-write. Reset: 0. This 32-bit data is used as the return data when the memory read trapping is enabled through <a href="#">SMIxD8[MemRdOvrEn]</a> .

**SMIxE0 MemTrapping2**

Bits	Description
31:2	<b>MemTrapping</b> . Read-write. Reset: 0. Specifies the 30-bit memory address which causes SMI event. The lowest 2 bits are ignored.

1	<b>MemRdOvrEn.</b> Read-write. Reset: 0. 1=Force read data to be replaced by <a href="#">SMIxE4 [MemRdOvrData2]</a> .
0	<b>MemTrappingRw.</b> Read-write. Reset: 0. 0=Trap on read access at the address specified by [Mem-Trapping]. 1=Trap on write access.

### SMIxE4 MemRdOvrData2

Bits	Description
31:0	<b>MemRdOvrData.</b> Read-write. Reset: 0. This 32-bit data is used as the return data when the memory read trapping is enabled through <a href="#">SMIxE0[MemRdOvrEn]</a> .

### SMIxE8 MemTrapping3

Bits	Description
31:2	<b>MemTrapping.</b> Read-write. Reset: 0. Specifies the 30-bit memory address which causes SMI event. The lowest 2 bits are ignored.
1	<b>MemRdOvrEn.</b> Read-write. Reset: 0. 1=Force read data to be replaced by <a href="#">SMIxEC [MemRdOvrData3]</a> .
0	<b>MemTrappingRw.</b> Read-write. Reset: 0. 0=Trap on read access at the address specified by [Mem-Trapping]. 1=Trap on write access.

### SMIxEC MemRdOvrData3

Bits	Description
31:0	<b>MemRdOvrData.</b> Read-write. Reset: 0. This 32-bit data is used as the return data when the memory read trapping is enabled through <a href="#">SMIxE8[MemRdOvrEn]</a> .

### SMIxF0 CfgTrapping0

Bits	Description
31:2	<b>CfgTrapping.</b> Read-write. Reset: 0. Specifies the AD[31:2] value during configure cycle which causes SMI event.
1	<b>IoTrappingRw0.</b> Read-write. Reset: 0. 0=Trap on IO read access on the address specified in <a href="#">SMIxC8 [IoTrapping0]</a> . 1=Trap on IO write access on the address specified in <a href="#">SMIxC8 [IoTrapping0]</a> .
0	<b>CfgTrappingRw.</b> Read-write. Reset: 0. 0=Trap on read access at the address specified in [CfgTrapping]. 1=Trap on write access.

### SMIxF4 CfgTrapping1

Bits	Description
31:2	<b>CfgTrapping.</b> Read-write. Reset: 0. Specifies the AD[31:2] value during configure cycle which causes SMI event.

1	<b>IoTrappingRw1.</b> Read-write. Reset: 0. 0=Trap on IO read access on the address specified in <a href="#">SMIxCA [IoTrapping1]</a> . 1=Trap on IO write access on the address specified in <a href="#">SMIxCA [IoTrapping1]</a> .
0	<b>CfgTrappingRw.</b> Read-write. Reset: 0. 0=Trap on read access at the address specified in [CfgTrapping]. 1=Trap on write access.

### SMIxF8 CfgTrapping2

Bits	Description
31:2	<b>CfgTrapping.</b> Read-write. Reset: 0. Specifies the AD[31:2] value during configure cycle which causes SMI event.
1	<b>IoTrappingRw2.</b> Read-write. Reset: 0. 0=Trap on IO read access on the address specified in <a href="#">SMIxCC [IoTrapping2]</a> . 1=Trap on IO write access on the address specified in <a href="#">SMIxCC [IoTrapping2]</a> .
0	<b>CfgTrappingRw.</b> Read-write. Reset: 0. 0=Trap on read access at the address specified in [CfgTrapping]. 1=Trap on write access.

### SMIxFC CfgTrapping3

Bits	Description
31:2	<b>CfgTrapping.</b> Read-write. Reset: 0. Specifies the AD[31:2] value during configure cycle which causes SMI event.
1	<b>IoTrappingRw3.</b> Read-write. Reset: 0. 0=Trap on IO read access at the address specified in <a href="#">SMIxCE [IoTrapping3]</a> . 1=Trap on IO write access at the address specified in <a href="#">SMIxCE [IoTrapping3]</a> .
0	<b>CfgTrappingRw.</b> Read-write. Reset: 0. 0=Trap on read access at the address specified in [CfgTrapping]. 1=Trap on write access.

### 3.25.17 Watchdog Timer (WDT) Registers

Watchdog timer registers are accessed through two methods:

- Memory access through watchdog timer base address defined by [PMx48 \[WatchdogTimerEn\]](#):
  1. Program the watchdog timer base address through [PMx48\[WatchdogBase\]](#).
  2. Enable watchdog timer address decoding through [PMx48\[WatchdogDecodeEn\]](#).
- Memory mapped or IO mapped access through the AcpiMmio region. The WDT registers start from [PMx24\[AcpiMmioAddr\]+B00h](#). See [PMx24 \[AcpiMmioEn\]](#).

#### WDTx00 WatchdogControl

Bits	Description
31:8	Reserved.
7	<b>WatchdogTrigger.</b> RAZ; write-1-only. Reset: 0. Writing 1 to this bit triggers the watchdog to start a new count interval, counting down from the value that was last written to <a href="#">WDTx04</a> . Setting this bit has no effect if the watchdog is disabled or stopped.
6:4	Reserved.
3	<b>WatchdogDisable.</b> Read-only. Reset: 1. This bit reflects the state of the watchdog timer hardware. 0=Enable. 1=Disable.
2	<b>WatchdogAction.</b> Read-write. Reset: 0. This bit determines the action to be taken when the watchdog timer expires. 0=System reset. 1=System power off. The bit is only valid when the watchdog is enabled.
1	<b>WatchdogFired.</b> Read; write-1-to-clear. Cold reset: 0. 1=The watchdog timer has expired and caused the current restart. The bit is cleared by a power cycle or by the operating system and it must remain cleared for any restart that is not caused by the watchdog timer firing. The bit is only valid when the watchdog is enabled.
0	<b>WatchdogRunStopB.</b> Read-write. Reset: 0. 1=Watchdog is in the running state. 0=Watchdog is in the stopped state. This bit is used to control or indicate whether the watchdog is in the running or stopped states. If the watchdog is in the stopped state and a 1 is written to this bit, the watchdog moves to the running state, but a count interval is not started until a 1 is written to bit [7]. If the watchdog is in the running state, writing 1 to this bit has no effect. The bit is only valid when the watchdog is enabled.

#### WDTx04 WatchdogCount

Bits	Description
31:16	Reserved.
15:0	<b>WatchdogCount.</b> Read-write. Reset: X. Writing this register specifies the countdown time for the counter. Reading this register returns the current counter value.

### 3.25.18 Wake Alarm Device (AcDcTimer) Registers

The AC/DC timer registers are used to control the wake alarm device. They are accessed through the AcpiMmio region, which can be memory mapped or IO mapped. The AC DC timer registers range from [PMx24\[AcpiMmioAddr\]+1D00h](#) to [PMx24\[AcpiMmioAddr\]+1DFFh](#). See [PMx24 \[AcpiMmioEn\]](#).

**AcDcTimerx00 AcTimerValue**

Bits	Description								
31:0	<p><b>AcTimerValue.</b> Read-write; updated-by-hardware. Reset: FFFF_FFFFh. Writing the register with a value other than FFFF_FFFFh starts the AC timer. Reading this register returns the current value of the AC timer.</p> <p>When the AC or DC timer generates a wake up event, this register is reset to FFFF_FFFFh by hardware.</p> <table> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>FFFF_FFFFh</td> <td>Disable the AC timer.</td> </tr> <tr> <td>FFFF_FFFEh-0000_0001h</td> <td>The value indicates the number of seconds between the time when the AC timer is programmed and the time when it expires.</td> </tr> <tr> <td>0000_0000h</td> <td>The AC timer wakes up the system instantly.</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	FFFF_FFFFh	Disable the AC timer.	FFFF_FFFEh-0000_0001h	The value indicates the number of seconds between the time when the AC timer is programmed and the time when it expires.	0000_0000h	The AC timer wakes up the system instantly.
<u>Bits</u>	<u>Definition</u>								
FFFF_FFFFh	Disable the AC timer.								
FFFF_FFFEh-0000_0001h	The value indicates the number of seconds between the time when the AC timer is programmed and the time when it expires.								
0000_0000h	The AC timer wakes up the system instantly.								

**AcDcTimerx04 AcExpiredTimerPolicy**

Bits	Description								
31:0	<p><b>AcExpiredTimerPolicy.</b> Read-write; updated-by-hardware. Reset: FFFF_FFFFh. When the AC timer expires, the wake signal is asserted if the current power source is AC; the wake signal is not asserted if the current power source is DC. If the power source is switched from DC to AC after the AC timer expired, we wait for the number of seconds defined in this register and then wake up the system. When the AC or DC timer generates a wake up event, this register is reset to FFFF_FFFFh by hardware.</p> <table> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>FFFF_FFFFh</td> <td>Disable the AC expired timer policy.</td> </tr> <tr> <td>FFFF_FFFEh-0000_0001h</td> <td>The value indicates the number of seconds between the time when the power is switched to AC and the time when it generates the wake-up event.</td> </tr> <tr> <td>0000_0000h</td> <td>The expired AC timer wakes up the system instantly once the power source is switched to AC.</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	FFFF_FFFFh	Disable the AC expired timer policy.	FFFF_FFFEh-0000_0001h	The value indicates the number of seconds between the time when the power is switched to AC and the time when it generates the wake-up event.	0000_0000h	The expired AC timer wakes up the system instantly once the power source is switched to AC.
<u>Bits</u>	<u>Definition</u>								
FFFF_FFFFh	Disable the AC expired timer policy.								
FFFF_FFFEh-0000_0001h	The value indicates the number of seconds between the time when the power is switched to AC and the time when it generates the wake-up event.								
0000_0000h	The expired AC timer wakes up the system instantly once the power source is switched to AC.								

**AcDcTimerx08 AcTimerStatus**

Bits	Description
31:2	Reserved.
1	<b>AcTimerWakeup.</b> Read; set-by-hardware; write-1-to-clear. Reset: 0. 1=Wake up was caused by AC timer expiration. 0=Wake-up was not caused by AC timer expiration.
0	<b>AcTimerExpired.</b> Read; set-by-hardware; write-1-to-clear. Reset: 0. 1=AC timer expired. 0=AC timer has not expired.



**AcDcTimerx10 DcTimerValue**

Bits	Description								
31:0	<p><b>DcTimerValue.</b> Read-write; updated-by-hardware. Reset: FFFF_FFFFh. Writing the register with a value other than FFFF_FFFFh starts the DC timer. Reading this register returns the current value of the DC timer.</p> <p>When the AC or DC Timer generates a wake up event, this register is reset to FFFF_FFFFh by hardware.</p> <table> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>FFFF_FFFFh</td> <td>Disable the DC timer.</td> </tr> <tr> <td>FFFF_FFFEh-0000_0001h</td> <td>The value indicates the number of seconds between the time when the DC timer is programmed and the time when it expires.</td> </tr> <tr> <td>0000_0000h</td> <td>The DC timer wakes up the system instantly.</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	FFFF_FFFFh	Disable the DC timer.	FFFF_FFFEh-0000_0001h	The value indicates the number of seconds between the time when the DC timer is programmed and the time when it expires.	0000_0000h	The DC timer wakes up the system instantly.
<u>Bits</u>	<u>Definition</u>								
FFFF_FFFFh	Disable the DC timer.								
FFFF_FFFEh-0000_0001h	The value indicates the number of seconds between the time when the DC timer is programmed and the time when it expires.								
0000_0000h	The DC timer wakes up the system instantly.								

**AcDcTimerx14 DcExpiredTimerPolicy**

Bits	Description								
31:0	<p><b>DcExpiredTimerPolicy.</b> Read-write; updated-by-hardware. Reset: FFFF_FFFFh. When the DC timer expires, the wake signal is asserted if the current power source is DC; the wake signal is not asserted if the current power source is AC. If the power source is switched from AC to DC after the DC timer expired, we wait for the number of seconds defined in this register and then wake up the system. When the AC or DC timer generates a wake up event, this register is reset to FFFF_FFFFh by hardware.</p> <table> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>FFFF_FFFFh</td> <td>Disable the DC expired timer policy.</td> </tr> <tr> <td>FFFF_FFFEh-0000_0001h</td> <td>The value indicates the number of seconds between the time when the power is switched to DC and the time when it generates the wake-up event.</td> </tr> <tr> <td>0000_0000h</td> <td>The expired DC timer wakes up the system instantly once the power source is switched to DC.</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	FFFF_FFFFh	Disable the DC expired timer policy.	FFFF_FFFEh-0000_0001h	The value indicates the number of seconds between the time when the power is switched to DC and the time when it generates the wake-up event.	0000_0000h	The expired DC timer wakes up the system instantly once the power source is switched to DC.
<u>Bits</u>	<u>Definition</u>								
FFFF_FFFFh	Disable the DC expired timer policy.								
FFFF_FFFEh-0000_0001h	The value indicates the number of seconds between the time when the power is switched to DC and the time when it generates the wake-up event.								
0000_0000h	The expired DC timer wakes up the system instantly once the power source is switched to DC.								

**AcDcTimerx18 DcTimerStatus**

Bits	Description
31:2	Reserved.
1	<b>DcTimerWakeup.</b> Read; set-by-hardware; write-1-to-clear. Reset: 0. 1=Wake-up was caused by DC timer expiration. 0=Wake-up was not caused by DC timer expiration.
0	<b>DcTimerExpired.</b> Read; set-by-hardware; write-1-to-clear. Reset: 0. 1=DC timer expired. 0=DC timer has not expired.

**AcDcTimerx20 AcDcTimerCtrl**

Bits	Description
31:10	Reserved.

9	<b>DcTimerEventEn.</b> Read-write. Reset: 0. 1=Enable DC Timer to wake up system. 0=Disable.
8	<b>AcTimerEventEn.</b> Read-write. Reset: 0. 1=Enable AC Timer to wake up system. 0=Disable.
7:1	Reserved.
0	<b>Busy.</b> Read-only; updated-by-hardware. Reset: 0h. Right after <a href="#">AcDcTimerx00 [AcTimerValue]</a> or <a href="#">AcDcTimerx10 [DcTimerValue]</a> is programmed, the hardware sets this bit to 1. The hardware will clear this bit once the programming is done and the corresponding timer is started properly. Before the software writes to <a href="#">AcDcTimerx00 [AcTimerValue]</a> or <a href="#">AcDcTimerx10 [DcTimerValue]</a> , it has to read this bit and make sure it is 0. Otherwise, the hardware will just ignore the programming action from software. For the registers other than <a href="#">AcDcTimerx00 [AcTimerValue]</a> or <a href="#">AcDcTimerx10 [DcTimerValue]</a> , there is no such limitation.

## 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.

45	SMMFEC0: SMM IO Trap Offset	201	D0F0x98_x4[A,9]: ORB LCLK Clock Control 1-0
46	SMMFEC4: Local SMI Status	201	D0F0xB8: SMU Index Address
46	SMMFEC8: SMM IO Restart Byte	201	D0F0xBC: SMU Index Data
47	SMMFEC9: Auto Halt Restart Offset	202	D0F0xBC_x3F800: FIRMWARE_FLAGS
47	SMMFECA: NMI Mask	202	D0F0xBC_x3F804: FIRMWARE_VID
47	SMMFED8: SMM SVM State	202	D0F0xBC_x3F820: PM_INTERVAL_CNTL_0
47	SMMFEFC: SMM-Revision Identifier	202	D0F0xBC_x3F828: PM_TIMER_PERIOD
48	SMMFF00: SMM Base Address (SMM_BASE)	202	D0F0xBC_x3F9D8: PM_CONFIG
115	MR0: DDR3 MR0	203	D0F0xBC_x3F9E8: NB_DPM_CONFIG_1
117	MR1: DDR3 MR1	204	D0F0xBC_x3F9EC: NB_DPM_CONFIG_2
117	MR2: DDR3 MR2	204	D0F0xBC_x3F9F4: CSR_GNB_1
118	MR3: DDR3 MR3	204	D0F0xBC_x3F9F8: CSR_GNB_3
187	IOCF8: IO-Space Configuration Address	204	D0F0xBC_x3FD[8C:00:step14]: LCLK DPM Control 0
188	IOCF8: IO-Space Configuration Data Port	205	D0F0xBC_x3FD[94:08:step14]: LCLK DPM Control 2
189	D0F0x00: Device/Vendor ID	205	D0F0xBC_x3FD[9C:10:step14]: LCLK DPM Activity Thresholds
189	D0F0x04: Status/Command	206	D0F0xBC_x3FDC8: SMU_LCLK_DPM_CNTL
189	D0F0x08: Class Code/Revision ID	206	D0F0xBC_x3FDD0:
189	D0F0x0C: Header Type	206	SMU_LCLK_DPM_THERMAL_THROTTLING_CNTL
190	D0F0x2C: Subsystem and Subvendor ID	206	D0F0xBC_x3FDD4:
190	D0F0x34: Capabilities Pointer	206	SMU_LCLK_DPM_THERMAL_THROTTLING_THRESHOLDS
190	D0F0x48: NB Header Write Register	206	D0F0xBC_xC020_008C: LCLK_DEEP_SLEEP_CNTL
190	D0F0x4C: PCI Control	207	D0F0xBC_xC020_0110: Activity Monitor Control
191	D0F0x60: Miscellaneous Index	207	D0F0xBC_xC210_0000: CPU Interrupt Request
191	D0F0x64: Miscellaneous Index Data	207	D0F0xBC_xC210_0004: CPU Interrupt Status
191	D0F0x64_x00: Northbridge Control	208	D0F0xBC_xC210_003C: CPU Interrupt Argument
191	D0F0x64_x0C: IOC Bridge Control	208	D0F0xBC_xC210_0040: CPU Interrupt Response
191	D0F0x64_x0D: IOC PCI Configuration	208	D0F0xC8: DEV Index Address
192	D0F0x64_x16: IOC Advanced Error Reporting Control	208	D0F0xCC: DEV Index Data
192	D0F0x64_x17: Memory Mapped IO Base Address	208	D0F0xCC_x01: IOC Bridge Control
192	D0F0x64_x18: Memory Mapped IO Limit	209	D0F0xD4_x0109_14C3: Bif Doorbell Control Ind
192	D0F0x64_x19: Top of Memory 2 Low	209	D0F0xD4_x0109_14E1: CC Bif Bx Strap0 Ind
192	D0F0x64_x1A: Top of Memory 2 High	209	D0F0xD4_x0109_14E2: CC Bif Bx Strap1 Ind
193	D0F0x64_x1D: Internal Graphics PCI Control	210	D0F0xD4_x0109_1507: CC Bif Bx Pinstrap0 Ind
193	D0F0x64_x1F: FCH Location	210	D0F0xE0: Link Index Address
193	D0F0x64_x22: LCLK Control 0	211	D0F0xE4: Link Index Data
194	D0F0x64_x23: LCLK Control 1	211	D0F0xE4_x0110_0010: PIF Control
194	D0F0x64_x3[4:0]: Programmable Device Remap Register	212	D0F0xE4_x0110_0011: PIF Pairing
195	D0F0x64_x46: IOC Features Control	212	D0F0xE4_x0110_001[8:7,3:2]: PIF Power Down Control [3:0]
195	D0F0x7C: IOC Configuration Control	214	D0F0xE4_x0120_0004: Phy Global Control 0
195	D0F0x84: Link Arbitration	214	D0F0xE4_x0120_4440: Phy RO PLL Control
196	D0F0x90: Northbridge Top of Memory	214	D0F0xE4_x0120_4450: PhyRO PLL Override Control 0
196	D0F0x94: Northbridge ORB Configuration Offset	215	D0F0xE4_x0130_0046: Subsystem and Vendor ID
196	D0F0x98: Northbridge ORB Configuration Data Port	215	D0F0xE4_x0130_0080: Link Configuration
196	D0F0x98_x06: ORB Downstream Control 0	215	D0F0xE4_x0130_0[C:8]00: Link Hold Training Control
197	D0F0x98_x07: ORB Upstream Arbitration Control 0	216	D0F0xE4_x0130_0[C:8]03: Link Deemphasis Control
197	D0F0x98_x08: ORB Upstream Arbitration Control 1	216	D0F0xE4_x0130_0[C:8]04: Link AER Control
198	D0F0x98_x09: ORB Upstream Arbitration Control 2	216	D0F0xE4_x0130_0[C:8]05: Link Training Control
198	D0F0x98_x0C: ORB Upstream Arbitration Control 5	217	D0F0xE4_x0130_8002: IO Link Wrapper Scratch
198	D0F0x98_x1E: ORB Receive Control 0	217	D0F0xE4_x0130_8011: Link Transmit Clock Gating Control
199	D0F0x98_x26: ORB IOMMU Control 0	217	D0F0xE4_x0130_8012: Link Idle-Resume Clock Gating Control
199	D0F0x98_x27: ORB IOMMU Control 1	218	D0F0xE4_x0130_8013: Transmit Clock Pll Control
199	D0F0x98_x28: ORB Transmit Control 0	218	D0F0xE4_x0130_8014: Link Transmit Clock Gating Control 2
199	D0F0x98_x2C: ORB Clock Control	219	D0F0xE4_x0130_8015: IO Link IOC Control
200	D0F0x98_x37: ORB Allow LDTSTOP Control 0	219	D0F0xE4_x0130_8016: Link Clock Switching Control
200	D0F0x98_x3A: ORB Source Tag Translation Control 2	220	D0F0xE4_x0130_802[4:1]: Transmitter Lane Mux
200	D0F0x98_x3B: ORB Source Tag Translation Control 3	221	D0F0xE4_x0130_802[8:5]: Receiver Lane Mux
		222	D0F0xE4_x0130_8029: Lane Enable
		222	D0F0xE4_x0130_8060: Soft Reset Command 0
		222	D0F0xE4_x0130_8062: Soft Reset Control 0
		223	D0F0xE4_x0130_80F0: BIOS Timer
		223	D0F0xE4_x0130_80F1: BIOS Timer Control
		223	D0F0xE4_x0140_0002: IO Link Hardware Debug
		223	D0F0xE4_x0140_0010: IO Link Control 1

224	D0F0xE4_x0140_0011: IO Link Config Control	245	D1F1x24: Base Address 5
224	D0F0xE4_x0140_001C: IO Link Control 2	245	D1F1x2C: Subsystem and Subvendor ID
225	D0F0xE4_x0140_0020: IO Link Chip Interface Control	245	D1F1x30: Expansion ROM Base Address
225	D0F0xE4_x0140_0040: IO Link Phy Control	246	D1F1x34: Capabilities Pointer
225	D0F0xE4_x0140_00B0: IO Link Strap Control	246	D1F1x3C: Interrupt Line
225	D0F0xE4_x0140_00C0: IO Link Strap Miscellaneous	246	D1F1x4C: Subsystem and Subvendor ID Mirror
226	D0F0xE4_x0140_00C1: IO Link Strap Miscellaneous2	246	D1F1x50: Power Management Capability
226	D0F0xF8: Northbridge IOAPIC Index	247	D1F1x54: Power Management Control and Status
226	D0F0xFC: Northbridge IOAPIC Data	247	D1F1x58: PCI Express® Capability
226	D0F0xFC_x00: IOAPIC Feature Control Register	247	D1F1x5C: Device Capability
226	D0F0xFC_x01: IOAPIC Base Address Lower	248	D1F1x60: Device Control and Status
227	D0F0xFC_x02: IOAPIC Base Address Upper	249	D1F1x64: Link Capability
227	D0F0xFC_x0F: IOAPIC GBIF Interrupt Routing Register	249	D1F1x68: Link Control and Status
227	D0F0xFC_x1[4:0]: IOAPIC BR Interrupt Routing Register	250	D1F1x7C: Device Capability 2
228	D0F0xFC_x30: IOAPIC Serial IRQ Status	250	D1F1x80: Device Control and Status 2
228	D0F0xFC_x3[F:E]: IOAPIC Scratch [1:0] Register	251	D1F1x84: Link Capability 2
229	D0F2xF8: PCIe® Client Interface Index	251	D1F1x88: Link Control and Status 2
229	D0F2xFC: PCIe® Client Interface Data	251	D1F1xA0: MSI Capability
229	D0F2xFC_x32_L1i: L1_CNTRL_4	252	D1F1xA4: MSI Message Address Low
230	D0F2xFC_x33_L1i: L1_CLKCNTRL_0	252	D1F1xA8: MSI Message Address High
231	D1F0x00: Device/Vendor ID	252	D1F1xAC: MSI Message Data
231	D1F0x04: Status/Command Register	252	D1F1x100: Vendor Specific Enhanced Capability
232	D1F0x08: Class Code/Revision ID Register	252	D1F1x104: Vendor Specific Header
232	D1F0x0C: Header Type Register	253	D1F1x108: Vendor Specific 1
232	D1F0x10: Graphic Memory Base Address	253	D1F1x10C: Vendor Specific 2
233	D1F0x14: Graphics Memory Base Address 64	254	D2F0x00: Device/Vendor ID (Host Bridge)
233	D1F0x18: Graphics Doorbell Base Address	254	D2F0x04: Status/Command
233	D1F0x1C: Graphics Doorbell Base Address 64	254	D2F0x08: Class Code/Revision ID
233	D1F0x20: Graphics IO Base Address	254	D2F0x0C: Header Type
234	D1F0x24: Graphics Memory Mapped Registers Base Address	254	D2F0x40: Header Type Write
234	D1F0x2C: Subsystem and Subvendor ID Register	255	D2F[5:1]x00: Device/Vendor ID
234	D1F0x30: Expansion ROM Base Address	255	D2F[5:1]x04: Status/Command Register
234	D1F0x34: Capabilities Pointer	256	D2F[5:1]x08: Class Code/Revision ID Register
234	D1F0x3C: Interrupt Line	256	D2F[5:1]x0C: Header Type Register
235	D1F0x4C: Subsystem and Subvendor ID Mirror	257	D2F[5:1]x18: Bus Number and Secondary Latency Register
235	D1F0x50: Power Management Capability	257	D2F[5:1]x1C: IO Base and Secondary Status Register
235	D1F0x54: Power Management Control and Status	258	D2F[5:1]x20: Memory Limit and Base Register
236	D1F0x58: PCI Express® Capability	258	D2F[5:1]x24: Prefetchable Memory Limit and Base Register
236	D1F0x5C: Device Capability	258	D2F[5:1]x28: Prefetchable Memory Base High Register
236	D1F0x60: Device Control and Status	258	D2F[5:1]x2C: Prefetchable Memory Limit High Register
237	D1F0x64: Link Capability	258	D2F[5:1]x30: IO Base and Limit High Register
238	D1F0x68: Link Control and Status	259	D2F[5:1]x34: Capabilities Pointer Register
239	D1F0x7C: Device Capability 2	259	D2F[5:1]x3C: Bridge Control Register
239	D1F0x80: Device Control and Status 2	259	D2F[5:1]x50: Power Management Capability Register
240	D1F0x84: Link Capability 2	260	D2F[5:1]x54: Power Management Control and Status Register
240	D1F0x88: Link Control and Status 2	260	D2F[5:1]x58: PCI Express® Capability Register
240	D1F0xA0: MSI Capability	261	D2F[5:1]x5C: Device Capability Register
241	D1F0xA4: MSI Message Address Low	261	D2F[5:1]x60: Device Control and Status Register
241	D1F0xA8: MSI Message Address High	262	D2F[5:1]x64: IO Link Capability Register
241	D1F0xAC: MSI Message Data	263	D2F[5:1]x68: IO Link Control and Status Register
241	D1F0x100: Vendor Specific Enhanced Capability	265	D2F[5:1]x6C: Slot Capability Register
242	D1F0x104: Vendor Specific Header	266	D2F[5:1]x70: Slot Control and Status Register
242	D1F0x108: Vendor Specific 1	266	D2F[5:1]x74: Root Complex Capability and Control Register
242	D1F0x10C: Vendor Specific 2	267	D2F[5:1]x78: Root Complex Status Register
243	D1F1x00: Device/Vendor ID	267	D2F[5:1]x7C: Device Capability 2
243	D1F1x04: Status/Command	268	D2F[5:1]x80: Device Control and Status 2
244	D1F1x08: Class Code/Revision ID	269	D2F[5:1]x84: IO Link Capability 2
244	D1F1x0C: Header Type	269	D2F[5:1]x88: IO Link Control and Status 2
244	D1F1x10: Audio Registers Base Address	270	D2F[5:1]x8C: Slot Capability 2
244	D1F1x14: Base Address 1	270	D2F[5:1]x90: Slot Control and Status 2
245	D1F1x18: Base Address 2	270	D2F[5:1]xA0: MSI Capability Register
245	D1F1x1C: Base Address 3	271	D2F[5:1]xA4: MSI Message Address Low
245	D1F1x20: Base Address 4	271	D2F[5:1]xA8: MSI Message Address High

271	D2F[5:1]xAC: MSI Message Data	297	D18F0x1DC: Core Enable
271	D2F[5:1]xB0: Subsystem and Subvendor Capability ID Register	299	D18F1x00: Device/Vendor ID
272	D2F[5:1]xB4: Subsystem and Subvendor ID Register	299	D18F1x08: Class Code/Revision ID
272	D2F[5:1]xB8: MSI Capability Mapping	299	D18F1x0C: Header Type
272	D2F[5:1]xBC: MSI Mapping Address Low	299	D18F1x[17C:140,7C:40]: DRAM Base/Limit
272	D2F[5:1]xC0: MSI Mapping Address High	301	D18F1x[2CC:2A0,1CC:180,BC:80]: MMIO Base/Limit
272	D2F[5:1]xE0: Root Port Index	303	D18F1x[DC:C0]: IO-Space Base/Limit
273	D2F[5:1]xE4: Root Port Data	305	D18F1x[1DC:1D0,EC:E0]: Configuration Map
273	D2F[5:1]xE4_x20: Root Port TX Control	306	D18F1xF0: DRAM Hole Address
273	D2F[5:1]xE4_x50: Root Port Lane Status	306	D18F1xF4: VGA Enable
273	D2F[5:1]xE4_x6A: Root Port Error Control	307	D18F1x10C: DCT Configuration Select
273	D2F[5:1]xE4_x70: Root Port Receiver Control	308	D18F1x120: DRAM Base System Address
274	D2F[5:1]xE4_xA0: Per Port Link Controller (LC) Control	308	D18F1x124: DRAM Limit System Address
274	D2F[5:1]xE4_xA1: LC Training Control	309	D18F1x2[1C:00]: DRAM Controller Base/Limit
275	D2F[5:1]xE4_xA2: LC Link Width Control	311	D18F2x00: Device/Vendor ID
275	D2F[5:1]xE4_xA3: LC Number of FTS Control	311	D18F2x08: Class Code/Revision ID
276	D2F[5:1]xE4_xA4: LC Link Speed Control	311	D18F2x0C: Header Type
276	D2F[5:1]xE4_xA5: LC State 0	311	D18F2x[5C:40]_dct[0]: DRAM CS Base Address
277	D2F[5:1]xE4_xB1: LC Control 2	313	D18F2x[6C:60]_dct[0]: DRAM CS Mask
277	D2F[5:1]xE4_xB5: LC Control 3	313	D18F2x78_dct[0]: DRAM Control
278	D2F[5:1]xE4_xC0: LC Strap Override	314	D18F2x7C_dct[0]: DRAM Initialization
278	D2F[5:1]xE4_xC1: Root Port Miscellaneous Strap Override	315	D18F2x80_dct[0]: DRAM Bank Address Mapping
279	D2F[5:1]xE4_xD0: Root Port ECC Skip OS Feature	316	D18F2x84_dct[0]: DRAM MRS
279	D2F[5:1]x100: Vendor Specific Enhanced Capability Register	317	D18F2x88_dct[0]: DRAM Timing Low
279	D2F[5:1]x104: Vendor Specific Header Register	317	D18F2x8C_dct[0]: DRAM Timing High
279	D2F[5:1]x108: Vendor Specific 1 Register	317	D18F2x90_dct[0]: DRAM Configuration Low
279	D2F[5:1]x10C: Vendor Specific 2 Register	319	D18F2x94_dct[0]: DRAM Configuration High
280	D2F[5:1]x128: Virtual Channel 0 Resource Status Register	322	D18F2x98_dct[0]: DRAM Controller Additional Data Offset
280	D2F[5:1]x150: Advanced Error Reporting Capability	322	D18F2x9C_dct[0]: DRAM Controller Additional Data Port
280	D2F[5:1]x154: Uncorrectable Error Status	323	D18F2x9C_x0000_0000_dct[0]_mp[1:0]: DRAM Output Driver Compensation Control
281	D2F[5:1]x158: Uncorrectable Error Mask	324	D18F2x9C_x0000_0[3:0]0[3:1]_dct[0]_mp[1:0]: DRAM Write Data Timing
282	D2F[5:1]x15C: Uncorrectable Error Severity	326	D18F2x9C_x0000_0004_dct[0]_mp[1:0]: DRAM Address/Command Timing Control
282	D2F[5:1]x160: Correctable Error Status	327	D18F2x9C_x0000_0[3:0]0[7:5]_dct[0]_mp[1:0]: DRAM Read DQS Timing
283	D2F[5:1]x164: Correctable Error Mask	328	D18F2x9C_x0000_0008_dct[0]_mp[1:0]: DRAM Phy Control
283	D2F[5:1]x168: Advanced Error Control	329	D18F2x9C_x0000_000B_dct[0]: DRAM Phy Status Register
284	D2F[5:1]x16C: Header Log DW0	329	D18F2x9C_x0000_000C_dct[0]: DRAM Phy Miscellaneous
284	D2F[5:1]x170: Header Log DW1	330	D18F2x9C_x0000_000D_dct[0]_mp[1:0]: DRAM Phy DLL Control
284	D2F[5:1]x174: Header Log DW2	331	D18F2x9C_x0000_00[2A:10]_dct[0]_mp[1:0]: DRAM DQS Receiver Enable Timing
284	D2F[5:1]x178: Header Log DW3	333	D18F2x9C_x0000_00[4A:30]_dct[0]_mp[1:0]: DRAM DQS Write Timing
284	D2F[5:1]x17C: Root Error Command	334	D18F2x9C_x0000_00[52:50]_dct[0]: DRAM Phase Recovery Control
285	D2F[5:1]x180: Root Error Status	335	D18F2x9C_x0D0F_0[F,8:0]02_dct[0]: Data Byte Transmit PreDriver Calibration
285	D2F[5:1]x184: Error Source ID	337	D18F2x9C_x0D0F_0[F,8:0]0[B,7,3]_dct[0]: Data Byte POdt Configuration
286	D18F0x00: Device/Vendor ID	337	D18F2x9C_x0D0F_0[F,8:0]04_dct[0]_mp[1:0]: Data Byte Pad Configuration
286	D18F0x04: Status/Command	338	D18F2x9C_x0D0F_0[F,8:0]0[A,6]_dct[0]: Data Byte Transmit PreDriver Calibration 2
286	D18F0x08: Class Code/Revision ID	338	D18F2x9C_x0D0F_0[F,8:0]10_dct[0]_mp[1:0]: Data Byte DLL Power Management
286	D18F0x0C: Header Type	339	D18F2x9C_x0D0F_0[F,8:0]13_dct[0]_mp[1:0]: Data Byte DLL Configuration
286	D18F0x34: Capabilities Pointer	340	D18F2x9C_x0D0F_0[F,8:0]1C_dct[0]_mp[1:0]: Data Byte DLL Power Management
286	D18F0x[5C:40]: Routing Table	341	D18F2x9C_x0D0F_0[F,8:0]1E_dct[0]_mp[1:0]: Data Byte Receiver Bias Current Control
287	D18F0x60: Node ID		
287	D18F0x64: Unit ID		
288	D18F0x68: Link Transaction Control		
289	D18F0x6C: Link Initialization Control		
291	D18F0x[E4,C4,A4,84]: Link Control		
291	D18F0x[EC,CC,AC,8C]: Link Feature Capability		
292	D18F0x[F0,D0,B0,90]: Link Base Channel Buffer Count		
294	D18F0x[F4,D4,B4,94]: Link Isochronous Channel Buffer Count		
295	D18F0x[F8,D8,B8,98]: Link Type		
295	D18F0x[11C,118,114,110]: Link Clumping Enable		
296	D18F0x150: Link Global Retry Control		
296	D18F0x168: Extended Link Transaction Control		
296	D18F0x16C: Link Global Extended Control		
297	D18F0x[18C:170]: Link Extended Control		
297	D18F0x1A0: Link Initialization Status		

341	D18F2x9C_x0D0F_0[F,8:0]1F_dct[0]_mp[1:0]: Data Byte Receiver Configuration	375	D18F2x224_dct[0]: DDR3 DRAM Timing 8
342	D18F2x9C_x0D0F_0[F,8:0]2[3:0]_dct[0]_mp[1:0]: Data Byte Rx Dqs DLL Configuration	376	D18F2x228_dct[0]: DDR3 DRAM Timing 9
344	D18F2x9C_x0D0F_0[8:0]2[9:4]_dct[0]_mp[1:0]: Data Byte DLL Configuration	376	D18F2x22C_dct[0]_mp[1:0]: DDR3 DRAM Timing 10
344	D18F2x9C_x0D0F_0[F,8:0]30_dct[0]: Data Byte DLL Configuration and Power Down	377	D18F2x[234:230]_dct[0]: DDR3 DRAM Read ODT Pattern [High:Low]
345	D18F2x9C_x0D0F_0[F,8:0]31_dct[0]: Data Byte Fence2 Threshold	377	D18F2x[23C:238]_dct[0]: DDR3 DRAM Write ODT Pattern [High:Low]
345	D18F2x9C_x0D0F_0[F,8:0]38_dct[0]: Data Byte DLL Control Register	378	D18F2x240_dct[0]_mp[1:0]: DDR3 DRAM ODT Control
346	D18F2x9C_x0D0F_1C00_dct[0]: Clock Transmit PreDriver Calibration	378	D18F2x244_dct[0]: DRAM Controller Miscellaneous 3
346	D18F2x9C_x0D0F_2[2:0]02_dct[0]: Clock Transmit PreDriver Calibration	379	D18F2x248_dct[0]_mp[1:0]: DRAM Power Management 0
347	D18F2x9C_x0D0F_[C,8,2][2:0]1E_dct[0]_mp[1:0]: Phy Control	380	D18F2x24C_dct[0]: DDR3 DRAM Power Management 1
347	D18F2x9C_x0D0F_[C,8,2][2:0]1F_dct[0]: Receiver Configuration	381	D18F2x250_dct[0]: DRAM Loopback and Training Control
348	D18F2x9C_x0D0F_[C,8,2][2:0]20_dct[0]_mp[1:0]: DLL Delay and Configuration	382	D18F2x25[8,4]_dct[0]: DRAM Target [B, A] Base
349	D18F2x9C_x0D0F_2[F,2:0]30_dct[0]: Clock DLL Configuration and Power Down	383	D18F2x25C_dct[0]: DRAM Command 0
349	D18F2x9C_x0D0F_4006_dct[0]: DRAM Phy MemVref Observation Configuration	383	D18F2x260_dct[0]: DRAM Command 1
349	D18F2x9C_x0D0F_4007_dct[0]: DRAM Phy MemVref Configuration	384	D18F2x264_dct[0]: DRAM Status 0
350	D18F2x9C_x0D0F_4009_dct[0]: Phy Cmp Configuration	384	D18F2x268_dct[0]: DRAM Status 1
350	D18F2x9C_x0D0F_[C,8][1:0]02_dct[0]: Transmit PreDriver Calibration	384	D18F2x26C_dct[0]: DRAM Status 2
351	D18F2x9C_x0D0F_[C,8][1:0][12,0E,0A,06]_dct[0]: Transmit PreDriver Calibration 2	385	D18F2x270_dct[0]: DRAM PRBS
351	D18F2x9C_x0D0F_812F_dct[0]: Tristate Configuration	385	D18F2x274_dct[0]: DRAM DQ Mask Low
352	D18F2x9C_x0D0F_[C,8][F:0]30_dct[0]: Cmd/Addr DLL Configuration and Power Down	385	D18F2x278_dct[0]: DRAM DQ Mask High
352	D18F2x9C_x0D0F_C021_dct[0]_mp[1:0]: DLL Delay and Configuration	385	D18F2x27C_dct[0]: DRAM ECC and EDC Mask
352	D18F2x9C_x0D0F_E000_dct[0]_mp[1:0]: Phy Master Configuration	386	D18F2x280_dct[0]: DRAM DQ Pattern Override 0
353	D18F2x9C_x0D0F_E006_dct[0]: Phy PLL Lock Time	386	D18F2x284_dct[0]: DRAM DQ Pattern Override 1
353	D18F2x9C_x0D0F_E007_dct[0]: Phy DLL Lock Time	386	D18F2x288_dct[0]: DRAM DQ Pattern Override 2
353	D18F2x9C_x0D00_E008_dct[0]: Phy Master Configuration	387	D18F2x28C_dct[0]: DRAM Command 2
354	D18F2x9C_x0D04_E008_dct[0]: Phy Master Configuration	388	D18F2x290_dct[0]: DRAM Status 3
354	D18F2x9C_x0D0F_E00A_dct[0]: Phy Dynamic Power Mode	388	D18F2x294_dct[0]: DRAM Status 4
354	D18F2x9C_x0D0F_E013_dct[0]: Phy PLL Regulator Wait Time	388	D18F2x298_dct[0]: DRAM Status 5
354	D18F2x9C_x0D0F_E019_dct[0]: Fence2	388	D18F2x29C_dct[0]: DRAM Status 6
355	D18F2x9C_x0D0F_E01A_dct[0]: M1 Fence Value	389	D18F2x2[B4,B0,AC,A8]_dct[0]: DRAM User Data Pattern
356	D18F2xA4: DRAM Controller Temperature Throttle	390	D18F2x2B8_dct[0]: DRAM Command 3
357	D18F2xA8_dct[0]: DRAM Controller Miscellaneous 2	390	D18F2x2[C0,BC]_dct[0]: DRAM Command 4 & 5
358	D18F2xAC: DRAM Controller Temperature Status	391	D18F2x2E0_dct[0]: Memory P-state Control and Status
358	D18F2xF8: P-state Power Information 1	391	D18F2x2E8_dct[0]_mp[1:0]: MRS Buffer
359	D18F2xFC: P-state Power Information 2	392	D18F2x2EC_dct[0]_mp[1:0]: MRS Buffer
359	D18F2x104: P-state Power Information 3	392	D18F2x2F0_dct[0]_mp[1:0]: DRAM Controller Misc 3
359	D18F2x110: DRAM Controller Select Low	392	D18F2x400_dct[0]: GMC to DCT Control 0
360	D18F2x114: DRAM Controller Select High	393	D18F2x404_dct[0]: GMC to DCT Control 1
361	D18F2x118: Memory Controller Configuration Low	393	D18F2x408_dct[0]: GMC to DCT Control 2
362	D18F2x11C: Memory Controller Configuration High	394	D18F2x420_dct[0]: GMC to DCT FIFO Config 1
365	D18F2x1B0: Extended Memory Controller Configuration Low	395	D18F3x00: Device/Vendor ID
366	D18F2x1B4: Extended Memory Controller Configuration High Register	395	D18F3x04: Status/Command
368	D18F2x1BC_dct[0]: DRAM CKE to CS Map	395	D18F3x08: Class Code/Revision ID
369	D18F2x200_dct[0]_mp[1:0]: DDR3 DRAM Timing 0	395	D18F3x0C: Header Type
369	D18F2x204_dct[0]_mp[1:0]: DDR3 DRAM Timing 1	395	D18F3x34: Capability Pointer
370	D18F2x208_dct[0]: DDR3 DRAM Timing 2	395	D18F3x40: MCA NB Control
371	D18F2x20C_dct[0]_mp[1:0]: DDR3 DRAM Timing 3	397	D18F3x44: MCA NB Configuration
371	D18F2x210_dct[0]_nbp[3:0]: DRAM NB P-state	400	D18F3x48: MCA NB Status Low
373	D18F2x214_dct[0]_mp[1:0]: DDR3 DRAM Timing 4	400	D18F3x4C: MCA NB Status High
373	D18F2x218_dct[0]_mp[1:0]: DDR3 DRAM Timing 5	400	D18F3x50: MCA NB Address Low
374	D18F2x21C_dct[0]_mp[1:0]: DDR3 DRAM Timing 6	400	D18F3x54: MCA NB Address High
375	D18F2x220_dct[0]: DDR3 DRAM Timing 7	400	D18F3x58: Scrub Rate Control
		401	D18F3x5C: DRAM Scrub Address Low
		402	D18F3x60: DRAM Scrub Address High
		402	D18F3x64: Hardware Thermal Control (HTC)
		403	D18F3x68: Software P-state Limit
		404	D18F3x6C: Data Buffer Count
		404	D18F3x70: SRI to XBAR Command Buffer Count
		405	D18F3x74: XBAR to SRI Command Buffer Count
		407	D18F3x78: MCT to XBAR Buffer Count
		408	D18F3x7C: Free List Buffer Count
		408	D18F3x[84:80]: ACPI Power State Control

411	D18F3x88: NB Configuration 1 Low (NB_CFG1_LO)	444	D18F5xE8: TDP Limit 3
411	D18F3x8C: NB Configuration 1 High (NB_CFG1_HI)	444	D18F5xEC: Load Step Throttle Control
411	D18F3xA0: Power Control Miscellaneous	444	D18F5x128: Clock Power/Timing Control 3
412	D18F3xA4: Reported Temperature Control	445	D18F5x12C: Clock Power/Timing Control 4
414	D18F3xA8: Pop Up and Down P-states	446	D18F5x16[C:0]: Northbridge P-state [3:0]
414	D18F3xB8: NB Array Address	448	D18F5x170: Northbridge P-state Control
414	D18F3xBC: NB Array Data Port	449	D18F5x174: Northbridge P-state Status
414	D18F3xBC_x8: DRAM ECC	450	D18F5x178: Northbridge Fusion Configuration
415	D18F3xD4: Clock Power/Timing Control 0	451	D18F5x17C: Miscellaneous Voltages
417	D18F3xD8: Clock Power/Timing Control 1	451	D18F5x188: Clock Power/Timing Control 5
418	D18F3xDC: Clock Power/Timing Control 2	452	D18F5x18C: Clock Power/Timing Control 6
419	D18F3xE4: Thermtrip Status	452	D18F5x194: Name String Address Port
420	D18F3xE8: Northbridge Capabilities	452	D18F5x198: Name String Data Port
421	D18F3xFC: CPUID Family/Model/Stepping	452	D18F5x198_x[B:0]: Name String Data
421	D18F3x138: DCT0 Bad Symbol Identification	453	D18F5x240: ECC Exclusion Base Address Low
421	D18F3x13C: DCT1 Bad Symbol Identification	453	D18F5x244: ECC Exclusion Base Address High
421	D18F3x140: SRI to XCS Token Count	453	D18F5x248: ECC Exclusion Limit Address Low
422	D18F3x144: MCT to XCS Token Count	453	D18F5x24C: ECC Exclusion Limit Address High
423	D18F3x1[54:48]: Link to XCS Token Count	454	D18F5x260: Clock Power/Timing Control 8
424	D18F3x160: NB Machine Check Misc (DRAM Thresholding) 0 (MC4_MISC0)	455	NBIOAPICx00: IO Register Select
424	D18F3x168: NB Machine Check Misc (Link Thresholding) 1 (MC4_MISC1)	455	NBIOAPICx10: IO Window
425	D18F3x17C: Extended Freelist Buffer Count	455	NBIOAPICx10_x00: IOAPIC ID
425	D18F3x180: Extended NB MCA Configuration	455	NBIOAPICx10_x01: IOAPIC Version
427	D18F3x188: NB Configuration 2	455	NBIOAPICx10_x02: IOAPIC Arbitration
427	D18F3x190: Downcore Control	456	NBIOAPICx10_x[4E:10:step2]: Redirection Table Entry [31:0]
428	D18F3x1A0: Core Interface Buffer Count	456	NBIOAPICx20: IRQ Pin Assertion
428	D18F3x1CC: IBS Control	456	NBIOAPICx40: EOI
429	D18F3x1FC: Product Information Register 1	457	APIC20: APIC ID
429	D18F3x200: Performance Mode Control Register	457	APIC30: APIC Version
429	D18F3x238: DCT2 Bad Symbol Identification	457	APIC80: Task Priority (TPR)
430	D18F3x23C: DCT3 Bad Symbol Identification	457	APIC90: Arbitration Priority (APR)
430	D18F3x2B4: DCT and Fuse Power Gate Control	458	APICA0: Processor Priority (PPR)
431	D18F4x00: Device/Vendor ID	458	APICB0: End of Interrupt
431	D18F4x04: Status/Command	458	APICC0: Remote Read
431	D18F4x08: Class Code/Revision ID	458	APICD0: Logical Destination (LDR)
431	D18F4x0C: Header Type	458	APICE0: Destination Format
431	D18F4x34: Capabilities Pointer	459	APICF0: Spurious-Interrupt Vector (SVR)
432	D18F4x110: Sample and Residency Timers	459	APIC[170:100]: In-Service (ISR)
432	D18F4x11[C:8]: C-state Control	459	APIC[1F0:180]: Trigger Mode (TMR)
435	D18F4x124: C-state Interrupt Control	460	APIC[270:200]: Interrupt Request (IRR)
435	D18F4x128: C-state Policy Control 1	460	APIC280: Error Status
436	D18F4x13C: SMU P-state Control	461	APIC300: Interrupt Command Low (ICR Low)
437	D18F4x15C: Core Performance Boost Control	462	APIC310: Interrupt Command High (ICR High)
437	D18F4x164: Fixed Errata	462	APIC320: LVT Timer
438	D18F4x16C: APM TDP Control	463	APIC330: LVT Thermal Sensor
438	D18F4x1C0: Node Cac Register 1	463	APIC340: LVT Performance Monitor
439	D18F4x250: TDP Limit 8	463	APIC3[60:50]: LVT LINT[1:0]
440	D18F5x00: Device/Vendor ID	464	APIC370: LVT Error
440	D18F5x04: Status/Command	464	APIC380: Timer Initial Count
440	D18F5x08: Class Code/Revision ID	464	APIC390: Timer Current Count
440	D18F5x0C: Header Type	464	APIC3E0: Timer Divide Configuration
440	D18F5x34: Capabilities Pointer	465	APIC400: Extended APIC Feature
440	D18F5x[70,60,50,40]: Northbridge Performance Event Select Low	465	APIC410: Extended APIC Control
441	D18F5x[74,64,54,44]: Northbridge Performance Event Select High	466	APIC420: Specific End Of Interrupt
441	D18F5x[78,68,58,48]: Northbridge Performance Event Counter Low	466	APIC[4F0:480]: Interrupt Enable
441	D18F5x[7C,6C,5C,4C]: Northbridge Performance Event Counter High	466	APIC[530:500]: Extended Interrupt [3:0] Local Vector Table
441	D18F5x80: Compute Unit Status 1	467	CPUID Fn0000_0000_EAX: Processor Vendor and Largest Standard Function Number
442	D18F5x84: Northbridge Capabilities 2	467	CPUID Fn0000_0000_E[D,C,B]X: Processor Vendor
443	D18F5x88: NB Configuration 4 (NB_CFG4)	467	CPUID Fn0000_0001_EAX: Family, Model, Stepping Identifiers
443	D18F5x8C: NB Configuration 5 (NB_CFG5)	468	CPUID Fn0000_0001_EBX: LocalApicId, LogicalProcessorCount, CLFlush
443	D18F5xE0: Processor TDP Running Average	468	CPUID Fn0000_0001_ECX: Feature Identifiers

469	CPUID Fn0000_0001_EDX: Feature Identifiers	484	CPUID Fn8000_0008_ECX: Size Identifiers
470	CPUID Fn0000_000[4:2]: Reserved	484	CPUID Fn8000_0008_EDX: Reserved
470	CPUID Fn0000_0005_EAX: Monitor/MWait	484	CPUID Fn8000_0009: Reserved
471	CPUID Fn0000_0005_EBX: Monitor/MWait	485	CPUID Fn8000_000A_EAX: SVM Revision and Feature Identification
471	CPUID Fn0000_0005_ECX: Monitor/MWait	485	CPUID Fn8000_000A_EBX: SVM Revision and Feature Identification
471	CPUID Fn0000_0005_EDX: Monitor/MWait	485	CPUID Fn8000_000A_ECX: SVM Revision and Feature Identification
471	CPUID Fn0000_0006_EAX: Thermal and Power Management	485	CPUID Fn8000_000A_EDX: SVM Revision and Feature Identification
471	CPUID Fn0000_0006_EBX: Thermal and Power Management	486	CPUID Fn8000_00[18:0B]: Reserved
471	CPUID Fn0000_0006_ECX: Thermal and Power Management	486	CPUID Fn8000_0019_EAX: L1 TLB 1G Identifiers
472	CPUID Fn0000_0006_EDX: Thermal and Power Management	486	CPUID Fn8000_0019_EBX: L2 TLB 1G Identifiers
472	CPUID Fn0000_0007_EAX_x0: Structured Extended Feature Identifiers (ECX=0)	487	CPUID Fn8000_0019_E[D,C]X: Reserved
472	CPUID Fn0000_0007_EBX_x0: Structured Extended Feature Identifiers (ECX=0)	487	CPUID Fn8000_001A_EAX: Performance Optimization Identifiers
472	CPUID Fn0000_0007_ECX_x0: Structured Extended Feature Identifiers (ECX=0)	487	CPUID Fn8000_001A_E[D,C,B]X: Reserved
473	CPUID Fn0000_0007_EDX_x0: Structured Extended Feature Identifiers (ECX=0)	488	CPUID Fn8000_001B_EAX: Instruction Based Sampling Identifiers
473	CPUID Fn0000_000[A:8]: Reserved	488	CPUID Fn8000_001B_E[D,C,B]X: Instruction Based Sampling Identifiers
473	CPUID Fn0000_000B: Reserved	488	CPUID Fn8000_001C_EAX: Lightweight Profiling Capabilities 0
473	CPUID Fn0000_000C: Reserved	488	CPUID Fn8000_001C_EBX: Lightweight Profiling Capabilities 0
473	CPUID Fn0000_000D_EAX_x0: Processor Extended State Enumeration (ECX=0)	488	CPUID Fn8000_001C_ECX: Lightweight Profiling Capabilities 0
473	CPUID Fn0000_000D_EBX_x0: Processor Extended State Enumeration (ECX=0)	489	CPUID Fn8000_001C_EDX: Lightweight Profiling Capabilities 0
474	CPUID Fn0000_000D_ECX_x0: Processor Extended State Enumeration (ECX=0)	489	CPUID Fn8000_001D_EAX_x0: Cache Properties
474	CPUID Fn0000_000D_EDX_x0: Processor Extended State Enumeration (ECX=0)	490	CPUID Fn8000_001D_EAX_x1: Cache Properties
474	CPUID Fn0000_000D_EAX_x1: Processor Extended State Enumeration (ECX=1)	490	CPUID Fn8000_001D_EAX_x2: Cache Properties
474	CPUID Fn0000_000D_E[D,C,B]X_x1: Processor Extended State Enumeration (ECX=1)	491	CPUID Fn8000_001D_EAX_x3: Cache Properties
474	CPUID Fn0000_000D_EAX_x2: Processor Extended State Enumeration (ECX=2)	491	CPUID Fn8000_001D_EBX_x0: Cache Properties
474	CPUID Fn0000_000D_EBX_x2: Processor Extended State Enumeration (ECX=2)	491	CPUID Fn8000_001D_EBX_x1: Cache Properties
475	CPUID Fn0000_000D_ECX_x2: Processor Extended State Enumeration (ECX=2)	491	CPUID Fn8000_001D_EBX_x2: Cache Properties
475	CPUID Fn0000_000D_EDX_x2: Processor Extended State Enumeration (ECX=2)	492	CPUID Fn8000_001D_EBX_x3: Cache Properties
475	CPUID Fn8000_0000_EAX: Largest Extended Function Number	492	CPUID Fn8000_001D_ECX_x0: Cache Properties
475	CPUID Fn8000_0000_E[D,C,B]X: Processor Vendor	492	CPUID Fn8000_001D_ECX_x1: Cache Properties
475	CPUID Fn8000_0001_EAX: Family, Model, Stepping Identifiers	492	CPUID Fn8000_001D_ECX_x2: Cache Properties
476	CPUID Fn8000_0001_EBX: BrandId Identifier	493	CPUID Fn8000_001D_ECX_x3: Cache Properties
476	CPUID Fn8000_0001_ECX: Feature Identifiers	493	CPUID Fn8000_001D_EDX_x0: Cache Properties
477	CPUID Fn8000_0001_EDX: Feature Identifiers	493	CPUID Fn8000_001D_EDX_x1: Cache Properties
478	CPUID Fn8000_000[4:2]_E[D,C,B,A]X: Processor Name String Identifier	493	CPUID Fn8000_001D_EDX_x2: Cache Properties
479	CPUID Fn8000_0005_EAX: L1 TLB 2M/4M Identifiers	494	CPUID Fn8000_001D_EDX_x3: Cache Properties
479	CPUID Fn8000_0005_EBX: L1 TLB 4K Identifiers	494	CPUID Fn8000_001E_EAX: Extended APIC ID
480	CPUID Fn8000_0005_ECX: L1 Data Cache Identifiers	494	CPUID Fn8000_001E_EBX: Compute Unit Identifiers
480	CPUID Fn8000_0005_EDX: L1 Instruction Cache Identifiers	494	CPUID Fn8000_001E_ECX: Node Identifiers
480	CPUID Fn8000_0006_EAX: L2 TLB 2M/4M Identifiers	495	CPUID Fn8000_001E_EDX: Reserved
481	CPUID Fn8000_0006_EBX: L2 TLB 4K Identifiers	495	MSR0000_0000: Load-Store MCA Address
481	CPUID Fn8000_0006_ECX: L2 Cache Identifiers	495	MSR0000_0001: Load-Store MCA Status
482	CPUID Fn8000_0006_EDX: L3 Cache Identifiers	495	MSR0000_0010: Time Stamp Counter (TSC)
482	CPUID Fn8000_0007_EAX: Processor Feedback Capabilities	495	MSR0000_001B: APIC Base Address (APIC_BAR)
482	CPUID Fn8000_0007_EBX: RAS Capabilities	496	MSR0000_002A: Cluster ID (EBL_CR_POWERON)
483	CPUID Fn8000_0007_ECX: Advanced Power Management Information	496	MSR0000_00E7: Max Performance Frequency Clock Count (MPERF)
483	CPUID Fn8000_0007_EDX: Advanced Power Management Information	496	MSR0000_00E8: Actual Performance Frequency Clock Count (APERF)
483	CPUID Fn8000_0008_EAX: Long Mode Address Size Identifiers	496	MSR0000_00FE: MTRR Capabilities (MTRRcap)
484	CPUID Fn8000_0008_EBX: Reserved	496	MSR0000_0174: SYSENTER CS (SYSENTER_CS)
		497	MSR0000_0175: SYSENTER ESP (SYSENTER_ESP)
		497	MSR0000_0176: SYSENTER EIP (SYSENTER_EIP)
		497	MSR0000_0179: Global Machine Check Capabilities (MCG_CAP)
		497	MSR0000_017A: Global Machine Check Status (MCG_STAT)
		498	MSR0000_017B: Global Machine Check Exception Reporting Control (MCG_CTL)
		498	MSR0000_01D9: Debug Control (DBG_CTL_MSR)
		498	MSR0000_01DB: Last Branch From IP (BR_FROM)
		498	MSR0000_01DC: Last Branch To IP (BR_TO)
		499	MSR0000_01DD: Last Exception From IP
		499	MSR0000_01DE: Last Exception To IP
		499	MSR0000_020[F:0]: Variable-Size MTRRs Base/Mask
		500	MSR0000_02[6F:68,59:58,50]: Fixed-Size MTRRs
		502	MSR0000_0277: Page Attribute Table (PAT)



503	MSR0000_02FF: MTRR Default Memory Type (MTRRdefType)	544	MSRC001_0049: FR Machine Check Control Mask (MC5_CTL_MASK)
503	MSR0000_0400: MC0 Machine Check Control (MC0_CTL)	544	MSRC001_00[53:50]: IO Trap (SMI_ON_IO_TRAP_[3:0])
504	MSR0000_0401: MC0 Machine Check Status (MC0_STATUS)	545	MSRC001_0054: IO Trap Control (SMI_ON_IO_TRAP_CTL_STS)
507	MSR0000_0402: MC0 Machine Check Address (MC0_ADDR)	546	MSRC001_0055: Interrupt Pending
507	MSR0000_0403: MC0 Machine Check Miscellaneous (MC0_MISC)	546	MSRC001_0056: SMI Trigger IO Cycle
508	MSR0000_0404: MC1 Machine Check Control (MC1_CTL)	546	MSRC001_0058: MMIO Configuration Base Address
508	MSR0000_0405: MC1 Machine Check Status (MC1_STATUS)	547	MSRC001_0060: BIST Results
510	MSR0000_0406: MC1 Machine Check Address (MC1_ADDR)	547	MSRC001_0061: P-state Current Limit
510	MSR0000_0407: MC1 Machine Check Miscellaneous (MC1_MISC)	548	MSRC001_0062: P-state Control
510	MSR0000_0408: MC2 Machine Check Control (MC2_CTL)	548	MSRC001_0063: P-state Status
511	MSR0000_0409: MC2 Machine Check Status (MC2_STATUS)	548	MSRC001_00[6B:64]: P-state [7:0]
514	MSR0000_040A: MC2 Machine Check Address (MC2_ADDR)	550	MSRC001_0070: COFVID Control
514	MSR0000_040B: MC2 Machine Check Miscellaneous (MC2_MISC)	551	MSRC001_0071: COFVID Status
514	MSR0000_040C: MC3 Machine Check Control (MC3_CTL)	552	MSRC001_0073: C-state Base Address
514	MSR0000_040D: MC3 Machine Check Status (MC3_STATUS)	552	MSRC001_0074: CPU Watchdog Timer (CpuWdtCfg)
514	MSR0000_040E: MC3 Machine Check Address (MC3_ADDR)	553	MSRC001_0080: Frequency Sensitivity Feedback Monitor Actual Count 0
515	MSR0000_040F: MC3 Machine Check Miscellaneous (MC3_MISC)	553	MSRC001_0081: Frequency Sensitivity Feedback Monitor Reference Count 0
515	MSR0000_0410: MC4 Machine Check Control (MC4_CTL)	553	MSRC001_0111: SMM Base Address (SMM_BASE)
516	MSR0000_0411: MC4 Machine Check Status (MC4_STATUS)	554	MSRC001_0112: SMM TSeg Base Address (SMMAddr)
520	MSR0000_0412: MC4 Machine Check Address (MC4_ADDR)	554	MSRC001_0113: SMM TSeg Mask (SMMMask)
523	MSR0000_0413: NB Machine Check Misc 4 (DRAM Thresholding) 0 (MC4_MISC0)	555	MSRC001_0114: Virtual Machine Control (VM_CR)
524	MSR0000_0414: MC5 Machine Check Control (MC5_CTL)	556	MSRC001_0115: IGNNE
524	MSR0000_0415: MC5 Machine Check Status (MC5_STATUS)	556	MSRC001_0116: SMM Control (SMM_CTL)
525	MSR0000_0416: MC5 Machine Check Address (MC5_ADDR)	557	MSRC001_0117: Virtual Machine Host Save Physical Address (VM_HSAVE_PA)
526	MSR0000_0417: MC5 Machine Check Miscellaneous (MC5_MISC)	557	MSRC001_0118: SVM Lock Key
527	MSRC000_0080: Extended Feature Enable (EFER)	557	MSRC001_011A: Local SMI Status
527	MSRC000_0081: SYSCALL Target Address (STAR)	557	MSRC001_0140: OS Visible Work-around MSR0 (OSVW_ID_Length)
527	MSRC000_0082: Long Mode SYSCALL Target Address (STAR64)	557	MSRC001_0141: OS Visible Work-around MSR1 (OSVW_Status)
528	MSRC000_0083: Compatibility Mode SYSCALL Target Address (STARCOMPAT)	558	MSRC001_023[6,4,2,0]: L2I Performance Event Select (L2I_PERF_CTL[3:0])
528	MSRC000_0084: SYSCALL Flag Mask (SYSCALL_FLAG_MASK)	559	MSRC001_023[7,5,3,1]: L2I Performance Event Counter (L2I_PERF_CTR[3:0])
528	MSRC000_0100: FS Base (FS_BASE)	559	MSRC001_024[6,4,2,0]: Northbridge Performance Event Select (NB_PERF_CTL[3:0])
528	MSRC000_0101: GS Base (GS_BASE)	560	MSRC001_024[7,5,3,1]: Northbridge Performance Event Counter (NB_PERF_CTR[3:0])
528	MSRC000_0102: Kernel GS Base (KernelGSbase)	561	MSRC001_1002: CPUID Features for CPUID Fn0000_0007_E[B,A]X_x0
528	MSRC000_0103: Auxiliary Time Stamp Counter (TSC_AUX)	561	MSRC001_1004: CPUID Features (Features)
529	MSRC000_0104: Time Stamp Counter Ratio (TscRateMsr)	563	MSRC001_1005: Extended CPUID Features (ExtFeatures)
529	MSRC000_0408: NB Machine Check Misc 4 (Link Thresholding) 1 (MC4_MISC1)	564	MSRC001_101[B:9]: Address Mask For DR[3:1] Breakpoints
531	MSRC000_0409: Reserved	565	MSRC001_1020: Load-Store Configuration (LS_CFG)
532	MSRC000_040[F:A]: Reserved	565	MSRC001_1021: Instruction Cache Configuration (IC_CFG)
533	MSRC001_00[03:00]: Performance Event Select (PERF_CTL[3:0])	566	MSRC001_1022: Data Cache Configuration (DC_CFG)
534	MSRC001_00[07:04]: Performance Event Counter (PERF_CTR[3:0])	566	MSRC001_1023: Bus Unit Configuration (BU_CFG)
535	MSRC001_0010: System Configuration (SYS_CFG)	566	MSRC001_1027: Address Mask For DR0 Breakpoints (DR0_ADDR_MASK)
535	MSRC001_0015: Hardware Configuration (HWCR)	566	MSRC001_1028: Floating Point Configuration (FP_CFG)
537	MSRC001_00[18,16]: IO Range Base (IORR_BASE[1:0])	567	MSRC001_102A: Bus Unit Configuration 2 (BU_CFG2)
538	MSRC001_00[19,17]: IO Range Mask (IORR_MASK[1:0])	567	MSRC001_1030: IBS Fetch Control (IbsFetchCtl)
538	MSRC001_001A: Top Of Memory (TOP_MEM)	568	MSRC001_1031: IBS Fetch Linear Address (IbsFetchLinAd)
538	MSRC001_001D: Top Of Memory 2 (TOM2)	568	MSRC001_1032: IBS Fetch Physical Address (IbsFetchPhysAd)
539	MSRC001_001F: Northbridge Configuration 1 (NB_CFG1)	568	MSRC001_1033: IBS Execution Control (IbsOpCtl)
540	MSRC001_0022: Machine Check Exception Redirection	569	MSRC001_1034: IBS Op Logical Address (IbsOpRip)
540	MSRC001_00[35:30]: Processor Name String	570	MSRC001_1035: IBS Op Data (IbsOpData)
541	MSRC001_003E: Hardware Thermal Control (HTC)	570	MSRC001_1036: IBS Op Data 2 (IbsOpData2)
541	MSRC001_0044: DC Machine Check Control Mask (MC0_CTL_MASK)	571	MSRC001_1037: IBS Op Data 3 (IbsOpData3)
542	MSRC001_0045: IC Machine Check Control Mask (MC1_CTL_MASK)	573	MSRC001_1038: IBS DC Linear Address (IbsDcLinAd)
542	MSRC001_0046: BU Machine Check Control Mask (MC2_CTL_MASK)	573	MSRC001_1039: IBS DC Physical Address (IbsDcPhysAd)
543	MSRC001_0047: Reserved (MC3_CTL_MASK)		
543	MSRC001_0048: NB Machine Check Control Mask (MC4_CTL_MASK)		

573	MSRC001_103A: IBS Control	588	PMCx0C7: Retired Branch Resyncs
574	MSRC001_103B: IBS Branch Target Address (BP_IBSTGT_RIP)	588	PMCx0C8: Retired Near Returns
574	MSRC001_1090: Processor Feedback Constants 0	588	PMCx0C9: Retired Near Returns Mispredicted
574	MSRC001_10A0: L2I Configuration (L2I_CFG)	588	PMCx0CA: Retired Mispredicted Taken Branch Instructions due to Target Mismatch
576	PMCx000: Dispatched FPU Operations	588	PMCx0CB: Retired MMX™/FP Instructions
576	PMCx001: FP Scheduler Empty	588	PMCx0CD: Interrupts-Masked Cycles
576	PMCx002: Dispatched Fast Flag FPU Operations	588	PMCx0CE: Interrupts-Masked Cycles with Interrupt Pending
576	PMCx003: Retired SSE/AVX Operations	589	PMCx0CF: Interrupts Taken
577	PMCx005: Retired Serializing Ops	589	PMCx0DB: FPU Exceptions
577	PMCx011: Retired x87 Floating Point Operations	589	PMCx0D[F:C]: DR[3:0] Breakpoint Matches
577	PMCx020: Segment Register Loads	589	PMCx1CF: Tagged IBS Ops
577	PMCx021: Pipeline Restart Due to Self-Modifying Code	589	PMCx1D0: Tagged IBS Ops Retired
578	PMCx022: Pipeline Restart Due to Probe Hit	590	L2IPMCx060: Command Related to Victim Buffers
578	PMCx024: Locked Operations	590	L2IPMCx061: Command Related to Masked Operations
578	PMCx026: Retired CLFLUSH Instructions	590	L2IPMCx062: Command Related to Read Block Operations
578	PMCx027: Retired CUID Instructions	590	L2IPMCx063: Command Related to Change to Dirty Operations
578	PMCx029: LS Dispatch	591	L2IPMCx06C: Response From System on Cache Refills
578	PMCx02A: Canceled Store to Load Forward Operations	591	L2IPMCx07D: Requests to L2 Cache
579	PMCx040: Data Cache Accesses	591	L2IPMCx07E: L2 Cache Misses
579	PMCx041: Data Cache Misses	591	L2IPMCx07F: L2 Fill/Writeback
579	PMCx042: Data Cache Refills from L2 or Northbridge	592	NBPMCx0E4: Memory Controller Bypass Counter Saturation
579	PMCx043: Data Cache Refills from the northbridge	592	NBPMCx0E8: Thermal Status
580	PMCx044: Data Cache Lines Evicted	593	NBPMCx0E9: CPU/IO Requests to Memory/IO
580	PMCx045: L1 DTLB Miss and L2 DTLB Hit	593	NBPMCx0EA: Cache Block Commands
580	PMCx046: DTLB Miss	594	NBPMCx0EB: Sized Commands
580	PMCx047: Misaligned Accesses	594	NBPMCx0EC: Probe Responses and Upstream Requests
580	PMCx04B: Prefetch Instructions Dispatched	595	NBPMCx1E0: CPU to DRAM Requests to Target Node
581	PMCx04C: DCACHE Misses by Locked Instructions	595	NBPMCx1E1: IO to DRAM Requests to Target Node
581	PMCx04D: L1 DTLB Hit	596	NBPMCx1E2: CPU Read Command Latency to Target Node 0-3
581	PMCx052: Ineffective Software Prefetches	596	NBPMCx1E3: CPU Read Command Requests to Target Node 0-3
581	PMCx054: Global Page Invalidations	596	NBPMCx1E4: CPU Read Command Latency to Target Node 4-7
581	PMCx062: Command Related to Read Block Operations	597	NBPMCx1E5: CPU Read Command Requests to Target Node 4-7
582	PMCx063: Command Related to Change to Dirty Operations	597	NBPMCx1E6: CPU Command Latency to Target Node 0-3/4-7
582	PMCx065: Memory Requests by Type	598	NBPMCx1E7: CPU Requests to Target Node 0-3/4-7
582	PMCx067: Data Cache Prefetches	598	NBPMCx1EB: Request Cache Status 1
583	PMCx068: MAB Requests	598	NBPMCx1F0: Memory Controller Requests
583	PMCx069: MAB Wait Cycles	599	NBPMCx3EC: DRAM Accesses
583	PMCx06C: System Response by Coherence State	599	NBPMCx3ED: DRAM Controller Page Table Overflows
584	PMCx06D: Data Written to System	600	NBPMCx3EE: Memory Controller DRAM Command Slots Missed
584	PMCx075: Cache Cross-invalidates	600	NBPMCx3EF: Memory Controller Turnarounds
584	PMCx076: CPU Clocks not Halted	601	IO000: Dma_Ch 0
585	PMCx162: PDC Miss	601	IO002: Dma_Ch 1
585	PMCx080: Instruction Cache Fetches	601	IO004: Dma_Ch 2
585	PMCx081: Instruction Cache Misses	601	IO006: Dma_Ch 3
585	PMCx082: Instruction Cache Refills from L2	601	IO008: Dma_Status
585	PMCx083: Instruction Cache Refills from System	601	IO009: Dma_WriteRequest
586	PMCx084: L1 ITLB Miss, L2 ITLB Hit	601	IO00A: Dma_WriteMask
586	PMCx085: ITLB Miss	602	IO00B: Dma_WriteMode
586	PMCx087: Instruction Fetch Stall	602	IO00C: Dma_Clear
586	PMCx088: Return Stack Hits	602	IO00D: Dma_MasterClr
586	PMCx089: Return Stack Overflows	602	IO00E: Dma_ClrMask
586	PMCx08B: Instruction Cache Victims	602	IO00F: Dma_AllMask
586	PMCx08C: Instruction Cache Lines Invalidated	602	IO020: IntrCntrl1Reg1
587	PMCx099: ITLB Reloads	602	IO021: IntrCntrl1Reg2
587	PMCx09A: ITLB Reloads Aborted	603	IO022: IMCR_Index
587	PMCx19A: Retired Indirect Branch Info	603	IO023: IMCR_Data
587	PMCx0C0: Retired Instructions	603	IO040: TimerCh0
587	PMCx0C1: Retired uops	603	IO041: TimerCh1
587	PMCx0C2: Retired Branch Instructions	603	IO042: TimerCh2
587	PMCx0C3: Retired Mispredicted Branch Instructions	603	IO043: Tmr1CntrlWord
587	PMCx0C4: Retired Taken Branch Instructions	604	IO060: Keyboard Data
588	PMCx0C5: Retired Taken Branch Instructions Mispredicted	604	IO061: Nmi Status
588	PMCx0C6: Retired Far Control Transfers		

604	IO064: Keyboard Status/Control	616	IOC01: Pci_Intr_Data
605	IO070: RtcAddrPort and NmiMask	616	IOC01_x0[7:0]: PCI INT[H#,G#,F#,E#,D#,C#,B#,A#] Map
605	IO071: RtcDataPort	617	IOC01_x08: Intr Misc Map
605	IO072: Alternate RTC AddrPort	617	IOC01_x09: Intr Misc 0 Map
605	IO073: Alternate RTC DataPort	618	IOC01_x0A: IntrMisc1Map
605	IO073_x00: RTC Seconds	618	IOC01_x0B: IntrMisc2Map
606	IO073_x01: RTC Seconds Alarm	618	IOC01_x0[7F:C]: PCI Interrupt Map
606	IO073_x02: RTC Minutes	618	IOC14: Pci_Error
606	IO073_x03: RTC Minutes Alarm	618	IOCD0: PM2_Index
606	IO073_x04: RTC Hours	619	IOCD1: PM2_Data
606	IO073_x05: RTC Hours Alarm	619	IOCD4: BIO SRAMIndex
607	IO073_x06: RTC Day of Week	619	IOCD5: BIO SRAM Data
607	IO073_x07: RTC Date of Month	619	IOCD6: PM_Index
607	IO073_x08: RTC Month	619	IOCD7: PM_Data
607	IO073_x09: RTC Year	619	IOCF9: System Reset Register
607	IO073_x0A: RTC Register A	621	ABx00: AB Index Register
608	IO073_x0B: RTC Register B	621	ABx04: AB Data Register
609	IO073_x0C: RTC Register C	621	ABx04_x54: Misc Control 1
609	IO073_x0D: RTC Date Alarm	622	ABx04_x58: B-Link RAB Control
609	IO073_x32: RTC AltCentury	622	ABx04_x80: B-Link DMA Prefetch Control
609	IO073_x48: RTC Century	622	ABx04_x90: BIF Control 0
610	IO073_x50: RTC Extended RAM Address Port	622	ABx04_x94: MSI Control
610	IO073_x53: RTC Extended RAM Data Port	623	ABx04_x204: SBG Upstream Control
610	IO073_x7E: RTC Time Clear	623	ABx04_x10054: A-Link Arbitration Control and Clock Control
610	IO073_x7F: RTC RAM Enable	623	ABx04_x10090: Misc Control 3
610	IO080: PostCode	624	D11F0x00: Device/Vendor ID
610	IO081: Dma_PageCh2	624	D11F0x04: Status/Command
611	IO082: Dma_PageCh3	625	D11F0x08: Revision ID/Class Code
611	IO083: Dma_PageCh1	626	D11F0x0C: Header Type Register
611	IO084: Dma_Page_Reserved1	626	D11F0x10: Primary IDE CS0 Base Address(BAR0)
611	IO085: Dma_Page_Reserved2	627	D11F0x14: Primary IDE CS1 Base Address(BAR1)
611	IO086: Dma_Page_Reserved3	627	D11F0x18: Secondary IDE CS0 Base Address (BAR2)
611	IO087: Dma_PageCh0	627	D11F0x1C: Secondary IDE CS1 Base Address (BAR3)
611	IO088: Dma_Page_Reserved4	627	D11F0x20: Bus Master Interface Register Base Address (BAR4)
611	IO089: Dma_PageCh6	628	D11F0x24: AHCI Base Address (BAR5)
612	IO08A: Dma_PageCh7	628	D11F0x2C: Subsystem ID and Subsystem Vendor ID
612	IO08B: Dma_PageCh5	628	D11F0x34: Capabilities Pointer
612	IO08C: Dma_Page_Reserved5	628	D11F0x3C: Interrupt Line
612	IO08D: Dma_Page_Reserved6	628	D11F0x40: Misc Control
612	IO08E: Dma_Page_Reserved7	629	D11F0x44: Watch Dog Control And Status
612	IO08F: Dma_Refresh	629	D11F0x50: MSI Control
612	IO092: FastInit	630	D11F0x54: MSI Address
613	IO0A0: IntrCntrl2Reg1	630	D11F0x58: MSI Upper Address
613	IO0A1: IntrCntrl2Reg2	630	D11F0x5C: MSI Data
613	IO0C0: Dma2_Ch4Addr	630	D11F0x60: Power Management Capability
613	IO0C2: Dma2_Ch4Cnt	631	D11F0x64: PCI Power Management Control And Status
613	IO0C4: Dma2_Ch5Addr	631	D11F0x70: Serial ATA Capability Register 0
613	IO0C6: Dma2_Ch5Cnt	632	D11F0x74: Serial ATA Capability Register 1
613	IO0C8: Dma2_Ch6Addr	632	D11F0x78: IDP Index Register
614	IO0CA: Dma2_Ch6Cnt	632	D11F0x7C: IDP Data Register
614	IO0CC: Dma2_Ch7Addr	632	D11F0x7C_x00: HBA Capabilities (CAP)
614	IO0CE: Dma_Ch7Cnt	632	D11F0x7C_x04: Global HBA Control (GHC)
614	IO0D0: Dma_Status	633	D11F0x7C_x08: Interrupt Status (IS)
614	IO0D2: Dma_WriteRequest	633	D11F0x7C_x0C: Ports Implemented (PI)
614	IO0D4: Dma_WriteMask	633	D11F0x7C_x10: AHCI Version (VS)
614	IO0D6: Dma_WriteMode	633	D11F0x7C_x14: Command Completion Coalescing Control (CCC_CTL)
614	IO0D8: Dma_Clear	633	D11F0x7C_x18: Command Completion Coalescing Ports (CCC_PORTS)
615	IO0DA: Dma_MasterClr	633	D11F0x7C_x1C: Enclosure Management Location (EM_LOC)
615	IO0DC: Dma_ClrMask	633	D11F0x7C_x20: Enclosure Management Control (EM_CTL)
615	IO0DE: Dma_AllMask	633	D11F0x7C_x24: HBA Capabilities Extended (CAP2)
615	IO0F0: NCP Error	633	D11F0x7C_x28: BIOS/OS Handoff Control and Status (BOHC)
615	IO4D0: IntrEdgeControl		
616	IOC00: Pci_Intr_Index		

634	D11F0x7C_x1[8,0]0: Port 1,0 Command List Base Address (PxCLB)	656	SATAx1[A,2]8: Port 1,0 Serial ATA Status (PxSSTS)
634	D11F0x7C_x1[8,0]4: Port 1,0 Command List Base Upper Address (PxCLBU)	657	SATAx1[A,2]C: Port 1,0 Serial ATA Control (PxSCTL)
634	D11F0x7C_x1[8,0]8: Port 1,0 FIS Base Address (PxFB)	658	SATAx1[B,3]0: Port 1,0 Serial ATA Error (PxSERR)
634	D11F0x7C_x1[8,0]C: Port 1,0 FIS Base Address Upper (PxFBU)	660	SATAx1[B,3]4: Port 1,0 Serial ATA Active (PxSACT)
634	D11F0x7C_x1[9,1]0: Port 1,0 Interrupt Status (PxIS)	661	SATAx1[B,3]8: Port 1,0 Command Issue (PxCI)
634	D11F0x7C_x1[9,1]4: Port 1,0 Interrupt Enable (PxIE)	661	SATAx1[B,3]C: Port 1,0 SNotification (PxSNTF)
634	D11F0x7C_x1[9,1]8: Port 1,0 Command and Status (PxCMD)	661	SATAx1[C,4]0: Port 1,0 FIS-based Switching Control (PxFBS)
635	D11F0x7C_x1[A,2]0: Port 1,0 Task File Data (PxTFD)	662	SATAx1[C,4]4: Port 1,0 Device Sleep (PxDEVSLP)
635	D11F0x7C_x1[A,2]4: Port 1,0 Signature (PxSIG)	663	SATA_EMx00: Message Header
635	D11F0x7C_x1[A,2]8: Port 1,0 Serial ATA Status (PxSSTS)	664	SATA_EMx04: Write SGPIO Register Request (I)
635	D11F0x7C_x1[A,2]C: Port 1,0 Serial ATA Control (PxSCTL)	664	SATA_EMx08: Write SGPIO Register Request (II)
635	D11F0x7C_x1[B,3]0: Port 1,0 Serial ATA Error (PxSERR)	665	SATA_EMx0C: Write SGPIO Register Request (III)
635	D11F0x7C_x1[B,3]4: Port 1,0 Serial ATA Active (PxSACT)	666	D[16,13,12]F0x00: Device/Vendor ID
635	D11F0x7C_x1[B,3]8: Port 1,0 Command Issue (PxCI)	666	D[16,13,12]F0x04: Status/Command
635	D11F0x7C_x1[B,3]C: Port 1,0 SNotification (PxSNTF)	667	D[16,13,12]F0x08: Class Code/Revision ID
636	D11F0x7C_x1[C,4]0: Port 1,0 FIS-based Switching Control (PxFBS)	667	D[16,13,12]F0x0C: Miscellaneous
636	D11F0x80: PHY Core Control 1	668	D[16,13,12]F0x10: OHCI Base Address
636	D11F0x84: PHY Core Control 2	668	D[16,13,12]F0x2C: Subsystem and Subvendor ID
637	D11F0x88: PHY Global Control 1	668	D[16,13,12]F0x34: Capabilities Pointer
637	D11F0x8C: PHY Global Control 2	668	D[16,13,12]F0x3C: Interrupt Line
637	D11F0x98: PHY Fine Tune PortX GenX Setting	668	D[16,13,12]F0x40: Miscellaneous 2
637	D11F0x9C: PortX Setting 2	669	D[16,13,12]F0x44: USB Battery Charger
638	D11F0xB4: PortX BIST Control/Status	669	D[16,13,12]F0x48: Port Force Reset
639	D11F0xB7: PortX BIST Port Select	669	D[16,13,12]F0x50: OHCI Misc Control
639	D11F0xBC: T-Mode BIST Transit Pattern DW1	670	D[16,13,12]F0x58: Over-Current Control 1
639	D11F0xC0: T-Mode BIST Transit Pattern DW2	671	D[16,13,12]F0x68: OHCI Over-Current PME Enable
639	D11F0xC4: T-Mode BIST Transit Control	671	D[16,13,12]F0x74: Target Timeout PME
639	D11F0xD0: Advanced Features Capability Register 0	671	D[16,13,12]F0x80: OHCI Spare 1
640	D11F0xD4: Advanced Features Capability Register 1	672	D[16,13,12]F0xD0: MSI Control
640	D11F0xE0: PCI Target Control TimeOut Counter	672	D[16,13,12]F0xD4: MSI Address
641	IDE[1:0]x00: IDE DATA	673	D[16,13,12]F0xD8: MSI Data
641	IDE[1:0]x01: Feature and Error	673	D[16,13,12]F0xE4: HT MSI Support
641	IDE[1:0]x02: Sector Count	673	D[16,13,12]F0xF0: Function Level Reset Capability
641	IDE[1:0]x03: Sector Number	673	D[16,13,12]F0xF4: Function Level Reset Control
641	IDE[1:0]x04: Cylinder Low	674	OHCI[3:1]x00: HC Revision
642	IDE[1:0]x05: Cylinder High	674	OHCI[3:1]x04: HC Control
642	IDE[1:0]x06: Drive and Head	675	OHCI[3:1]x08: HC Command Status
642	IDE[1:0]x07: Command and Status	675	OHCI[3:1]x0C: HC Interrupt Status
642	IDEA[1:0]x02: Device Control and Alternate Status	676	OHCI[3:1]x10: HC Interrupt Enable
642	IDE_BMx0[8,0]: Bus Master IDE Command	676	OHCI[3:1]x14: HC Interrupt Disable
643	IDE_BMx0[A,2]: Bus-master IDE Status	677	OHCI[3:1]x18: HC HCCA
643	IDE_BMx0[C,4]: Descriptor Table Pointer	677	OHCI[3:1]x1C: HC Period Current ED
644	SATAx00: HBA Capabilities (CAP)	677	OHCI[3:1]x20: HC Control Head ED
645	SATAx04: Global HBA Control (GHC)	677	OHCI[3:1]x24: HC Control Current ED
646	SATAx08: Interrupt Status (IS)	677	OHCI[3:1]x28: HC Bulk Head ED
647	SATAx0C: Ports Implemented (PI)	678	OHCI[3:1]x2C: HC BulkCurrent ED
647	SATAx10: AHCI Version (VS)	678	OHCI[3:1]x30: HC Done Head
647	SATAx14: Command Completion Coalescing Control (CCC_CTL)	678	OHCI[3:1]x34: HC Frame Interval
648	SATAx18: Command Completion Coalescing Ports (CCC_PORTS)	678	OHCI[3:1]x38: HC Frame Remaining
648	SATAx1C: Enclosure Management Location (EM_LOC)	679	OHCI[3:1]x3C: HC Frame Number
648	SATAx20: Enclosure Management Control (EM_CTL)	679	OHCI[3:1]x40: HC Periodic Start
649	SATAx24: HBA Capabilities Extended (CAP2)	679	OHCI[3:1]x44: HC LS Threshold
649	SATAx28: BIOS/OS Handoff Control and Status (BOHC)	679	OHCI[3:1]x48: HC RH Descriptor A
650	SATAx1[8,0]0: Port 1,0 Command List Base Address (PxCLB)	680	OHCI[3:1]x4C: HC RH Descriptor B
650	SATAx1[8,0]4: Port 1,0 Command List Base Upper Address (PxCLBU)	681	OHCI[3:1]x50: HC RH Status
650	SATAx1[8,0]8: Port 1,0 FIS Base Address (PxFB)	681	OHCI[3:1]x[60:54:step4]: HC RH Port Status [4:1]
650	SATAx1[8,0]C: Port 1,0 FIS Base Address Upper (PxFBU)	683	OHCI[3:1]xF0: HC Loopback Control
651	SATAx1[9,1]0: Port 1,0 Interrupt Status (PxIS)	684	OHCI[3:1]x100: HCE Control
652	SATAx1[9,1]4: Port 1,0 Interrupt Enable (PxIE)	685	OHCI[3:1]x104: HCE Input
653	SATAx1[9,1]8: Port 1,0 Command and Status (PxCMD)	685	OHCI[3:1]x108: HCE Output
655	SATAx1[A,2]0: Port 1,0 Task File Data (PxTFD)	685	OHCI[3:1]x10C: HCE Status
656	SATAx1[A,2]4: Port 1,0 Signature (PxSIG)	686	D[16,13,12]F2x00: Device/Vendor ID
		686	D[16,13,12]F2x04: Status/Command

687	D[16,13,12]F2x08: Revision ID / Class Code	715	D10F0x4C_x4000_000C: USB Common PHY Control
687	D[16,13,12]F2x0C: Miscellaneous	716	D10F0x4C_x4000_0010: HS Loopback Test
687	D[16,13,12]F2x10: BAR_EHCI	716	D10F0x4C_x4000_0014: CL Loopback Control
688	D[16,13,12]F2x2C: Subsystem ID / Subsystem Vendor ID	717	D10F0x4C_x4000_0018: Misc Control
688	D[16,13,12]F2x34: Capability Pointer	717	D10F0x4C_x4000_0020: USB SCLK Event Counter 0
688	D[16,13,12]F2x3C: Interrupt Line	717	D10F0x4C_x4000_0024: USB SCLK Event Counter 1
688	D[16,13,12]F2x50: EHCI Misc Control	717	D10F0x4C_x4000_0028: USB SCLK Event Counter Select
690	D[16,13,12]F2x54: EHCI Spare 1	718	D10F0x4C_x4000_002C: USB SCLK Event Counter Control
691	D[16,13,12]F2x60: Serial Bus Release Number / FLADJ	718	D10F0x4C_x4000_0048: LPM Control
691	D[16,13,12]F2xA0: USB Legacy Support Extended Capability	719	D10F0x4C_x4000_0050: USB Common PHY Control 2
691	D[16,13,12]F2xA4: USB Legacy Support Control / Status	719	D10F0x4C_x4000_0054: USB Common PHY Control 3
692	D[16,13,12]F2xC0: PME Capability	719	D10F0x50: PME Capability
693	D[16,13,12]F2xC4: PME Control / Status	720	D10F0x54: PME Control / Status
693	D[16,13,12]F2xD0: MSI Control	721	D10F0x60: SBRN
693	D[16,13,12]F2xD4: MSI Address	721	D10F0x70: MSI Control
694	D[16,13,12]F2xD8: MSI Upper Address	722	D10F0x74: MSI Address
694	D[16,13,12]F2xDC: MSI Data	722	D10F0x78: MSI Upper Address
694	D[16,13,12]F2xF0: Function Level Reset Capability	722	D10F0x7C: MSI Data
694	D[16,13,12]F2xF4: Function Level Reset Control	722	D10F0x80: MSI Mask Bits
695	EHCI[3:1]x00: Capability Length	722	D10F0x90: MSI-X Control
695	EHCI[3:1]x02: HC Interface Version	723	D10F0x94: MSI-X Table Offset/Table BIR
695	EHCI[3:1]x04: HC Structural Parameters	723	D10F0x98: MSI-X PBA Offset/PBA BIR
696	EHCI[3:1]x08: HC Capability Parameters	723	D10F0xA0: PCIe® Capability List
697	EHCI[3:1]x20: USB Command	724	D10F0xA4: Device Capability
699	EHCI[3:1]x24: USB Status	725	D10F0xA8: Device Control/Status
700	EHCI[3:1]x28: USB Interrupt Enable	726	D10F0xC4: Device Capabilities 2
701	EHCI[3:1]x2C: Frame Index	726	D10F0xC8: Device Control/Status 2
701	EHCI[3:1]x30: Control Data Structure Segment	727	D10F0x100: LTR Extended Capability Header
701	EHCI[3:1]x34: Periodic Frame List Base Address	727	D10F0x104: Max Latency
701	EHCI[3:1]x38: Current Async List Address	727	XHCI_CAPx02: HC Interface Version
702	EHCI[3:1]x60: Configure Flag	728	XHCI_PMx00: xHCI Config 0
702	EHCI[3:1]x[70,6C,68,64]: Port Status Control [4:1]	729	XHCI_PMx04: xHCI Firmware Addr 0
704	EHCI[3:1]xA4: Packet Buffer Threshold Values	729	XHCI_PMx08: xHCI Firmware Addr 1
705	EHCI[3:1]xB4: UTMI Control	729	XHCI_PMx10: xHCI Memory Config
705	EHCI[3:1]xB8: Loopback Test	730	XHCI_PMx14: PLL Control
706	EHCI[3:1]xBC: EOR MISC Control	730	XHCI_PMx20: USB2.0 Wake Control
707	EHCI[3:1]xC0: USB Common PHY CAL and Control	730	XHCI_PMx30: xHCI 1.0 Enable
707	EHCI[3:1]xC4: USB Common PHY Control 1	731	XHCI_PMx48: SSPHY ACPI Indirect Index
708	EHCI[3:1]xD0: USB Common PHY Control 2	731	XHCI_PMx4C: SSPHY ACPI IndirectData
708	EHCI[3:1]xD4: USB Common PHY Control 3	731	XHCI_PMx8C: SSPHY Common Control 0
708	EHCI[3:1]xDC: USB Battery Charger Enable	732	XHCI_PMxA0: SPI BAR0
709	D10F0x00: Device/Vendor ID	732	XHCI_PMxA4: SPI BAR1
709	D10F0x04: Status/Command	732	XHCI_PMxA8: SPI BAR2
710	D10F0x08: Revision ID / Class Code	732	XHCI_PMxAC: SPI BAR3
710	D10F0x0C: Miscellaneous	732	XHCI_PMxB0: SPI Valid Base
710	D10F0x10: Bar 0	733	XHCI_PMxB4: SPI Misc
711	D10F0x14: Bar 1	733	XHCI_PMx[FC:C0]: SPI Data Block N
711	D10F0x2C: Subsystem Vendor ID / Subsystem ID	734	D14F2x00: Device/Vendor ID
711	D10F0x34: Capability Pointer	734	D14F2x04: Status/Command
711	D10F0x3C: Interrupt Line	734	D14F2x08: Class Code/Revision ID
711	D10F0x40: IDP Index Register	735	D14F2x0C: Header Type
712	D10F0x44: IDP Data Register	735	D14F2x10: Lower Base Address Register
712	D10F0x48: Indirect PCI Index Register	735	D14F2x14: Upper Base Address Register
712	D10F0x4C: Indirect PCI Data Register	735	D14F2x2C: Subsystem and Subvendor ID
712	D10F0x4C_x08: Port Disable Write Once	735	D14F2x34: Capabilities Pointer
713	D10F0x4C_x0C: Port Disable RW	736	D14F2x3C: Interrupt Line
713	D10F0x4C_x10: USB DCLK Event Counter 0	736	D14F2x40: Misc Control
713	D10F0x4C_x14: USB DCLK Event Counter 1	737	D14F2x44: Interrupt Pin/Capability
713	D10F0x4C_x18: USB DCLK Event Counter Select	737	D14F2x50: Power Management Capability
714	D10F0x4C_x1C: USB DCLK Event Counter Control	738	D14F2x54: Power Management Control and Status
714	D10F0x4C_x4000_0000: UTMI Control	738	D14F2x60: MSI Capability
714	D10F0x4C_x4000_0004: USB PHY Status	738	D14F2x64: MSI Message Address Low
715	D10F0x4C_x4000_0008: USB Common PHY Calibration and Control	739	D14F2x68: MSI Message Address High

739	D14F2x6C: MSI Message Data	766	ASFx04: SlaveAddress
739	D14F2x70: Advanced Features Capability	766	ASFx05: Data0
739	D14F2x74: Advanced Features Control and Status	766	ASFx06: Data1
741	D14F7x00: Device/Vendor ID	767	ASFx07: DataIndex
741	D14F7x04: Status/Command	767	ASFx08: PEC
742	D14F7x08: Revision ID/Class Code	767	ASFx09: ListenAdr
742	D14F7x0C: Cache Line Size	767	ASFx0A: ASFStatus
742	D14F7x10: Base Address Reg 0	767	ASFx0B: StatusMask0
742	D14F7x14: Upper Base Address Reg 0	768	ASFx0C: StatusMask1
743	D14F7x2C: Subsystem ID and Subsystem Vendor ID	768	ASFx0D: SlaveStatus
743	D14F7x34: Capabilities Pointer	768	ASFx0E: RemoteCtrlAdr
743	D14F7x3C: Interrupt Line	768	ASFx0F: SensorAdr
743	D14F7x40: Slot Information	768	ASFx10: DataReadPointer
744	D14F7x80: SD PCI MSI Capability Header	768	ASFx11: DataWritePointer
744	D14F7x84: SD PCI MSI Address	769	ASFx12: SetDataReadPointer
744	D14F7x88: SD PCI MSI Upper Address	769	ASFx13: DataBankSel
744	D14F7x8C: SD PCI MSI Data	769	ASFx14: Semaphore
745	D14F7x90: Power Management Capability Header	770	ASFx15: SlaveEn
745	D14F7x94: Power Management Control and Status Register	770	ASFx16: DelayMasterTimer
746	D14F7xB8: SD Auto Pattern	770	SMBUSx00: SMBusStatus
746	SDHCx00: SDHC System Address / Argument 2	771	SMBUSx01: SMBusSlaveStatus
746	SDHCx04: SDHC Block CS	771	SMBUSx02: SMBusControl
747	SDHCx08: SDHC Command Argument	771	SMBUSx03: SMBusHostCmd
747	SDHCx0C: SDHC Command/Transfer Mode	772	SMBUSx04: SMBusAddress
748	SDHCx10: SDHC_RESP1_0	772	SMBUSx05: SMBusData0
748	SDHCx14: SDHC_RESP3_2	772	SMBUSx06: SMBusData1
748	SDHCx18: SDHC_RESP5_4	772	SMBUSx07: SMBusBlockData
748	SDHCx1C: SDHC_RESP7_6	772	SMBUSx08: SMBusSlaveControl
749	SDHCx20: SDHC_BUFFER	773	SMBUSx09: SMBusShadowCmd
749	SDHCx24: SDHC_PRSNT_STATE	773	SMBUSx0A: SMBusSlaveEvent
750	SDHCx28: SDHC_CTRL1	773	SMBUSx0C: SlaveData
751	SDHCx2C: SDHC_CTRL2	774	SMBUSx0E: SMBusTiming
752	SDHCx30: SDHC_INT_STATUS	774	SMBUSx10: I2CbusConfig
753	SDHCx34: SDHC_INT_MASK	774	SMBUSx11: I2CCommand
754	SDHCx38: SDHC_SIG_MASK	774	SMBUSx12: I2Cshadow1
754	SDHCx3C: SDHC ACMD12 Error/Host Control 2	774	SMBUSx13: I2Cshadow2
755	SDHCx40: SDHC_CAPABILITY	775	SMBUSx14: SMBusAutoPoll
756	SDHCx44: SDHC_CAPABILITY 2	775	SMBUSx15: SMBusCounter
757	SDHCx48: SDHC_CURR_CAPABILITY	775	SMBUSx16: SMBusStop
757	SDHCx50: SDHC_FORCE_EVT	776	SMBUSx17: SMBusHostCmd2
758	SDHCx54: SDHC_ADMA_ERR	777	IOAPICx00: IO Register Select Register
758	SDHCx58: SDHC_ADMA_SAD	777	IOAPICx10: IO Window Register
758	SDHCx6[C:0:step4]: Preset Value	777	IOAPICx10_x00: IOAPIC ID Register
759	SDHCxE0: Shared Bus Control Register	777	IOAPICx10_x01: IOAPIC Version Register
760	SDHCxFC: SDHC_VER_SLOT	777	IOAPICx10_x02: IOAPIC Arbitration Register
762	D14F0x00: Device/Vendor ID	778	IOAPICx10_x[3E:10:step2]: Redirection Table Entry [23:0]
762	D14F0x04: Status/Command	778	IOAPICx20: IRQ Pin Assertion Register
763	D14F0x08: Revision ID/Class Code	778	IOAPICx40: EOI Register
763	D14F0x0C: Cache Line Size	779	D14F3x00: Device/Vendor ID
763	D14F0x10: Base Address 0	779	D14F3x04: Status/Command
763	D14F0x14: Base Address 1	780	D14F3x08: Revision ID/Class Code
764	D14F0x18: Base Address 2	780	D14F3x0C: Cache Line Size
764	D14F0x1C: Base Address 3	780	D14F3x10: Base Address Reg 0
764	D14F0x20: Base Address 4	780	D14F3x2C: Subsystem ID and Subsystem Vendor ID
764	D14F0x24: Base Address 5	780	D14F3x34: Capabilities Pointer
764	D14F0x28: Cardbus CIS Pointer	781	D14F3x40: PCI Control
764	D14F0x2C: Subsystem Vendor ID	781	D14F3x44: IO Port Decode Enable
764	D14F0x30: Expansion ROM Base Address	783	D14F3x48: IO/Mem Port Decode Enable
765	D14F0x34: Capability Pointer	784	D14F3x4C: Memory Range
765	D14F0x3C: Interrupt Line	784	D14F3x[5C,58,54,50]: ROM Protect 3, 2, 1, 0
765	ASFx00: HostStatus	784	D14F3x60: PCI Memory Address for LPC Target Cycles
766	ASFx02: HostControl	785	D14F3x64: PCI IO base Address for Wide Generic Port
766	ASFx03: HostCommand	785	D14F3x68: ROM Address Range 1

785	D14F3x6C: ROM Address Range 2	810	MISCx10: CGPLLConfig3
786	D14F3x74: Alternative Wide IO Range Enable	810	MISCx14: CGPLLConfig4
786	D14F3x78: Miscellaneous Control Bits	811	MISCx18: CGPLLConfig5
786	D14F3x7C: TPM	811	MISCx1C: CGPLLConfig6
787	D14F3x84: TMKBC_BaseAddrLow	812	MISCx20: IMPCalibration
788	D14F3x88: TMKBC_BaseAddrHigh	813	MISCx24: ClkDrvStr1
788	D14F3x8C: TMKBC_Remap	815	MISCx28: ClkDrvStr2
788	D14F3x90: Wide IO 2	816	MISCx2C: ClkGatedCntl
788	D14F3x98: EC_LPC_Cntrl	817	MISCx30: CGPLLConfig7
788	D14F3xA0: SPI_Base_Addr	818	MISCx34: CGPLLConfig8
789	D14F3xA4: EC_PortAddress	818	MISCx38: CGPLLConfig9
789	D14F3xB0: RomDmaSrcAddr	818	MISCx3C: CGPLLConfig10
789	D14F3xB4: RomDmaDstAddr	819	MISCx40: MiscClkCntl1
789	D14F3xB8: RomDmaControl/EcControl/HostControl	820	MISCx44: MiscClkCntl2
790	D14F3xC0: EcRomWrOffset	821	MISCx48: MiscClkCntl3
791	D14F3xC4: EcRomRdOffset	822	MISCx4C: MiscClkCntl4
791	D14F3xC8: ClientRomProtect	822	MISCx60: IdleCntrl
791	D14F3xCC: AutoRomCfgr	823	MISCx68: Memory Power Saving Control
791	D14F3xD0: ClkCntrl	823	MISCx70: OscFreqCounter
792	D14F3xD4: ClkRunOption	823	MISCx74: HpetClkPeriod
792	SPIx00: SPI_Cntrl0	824	MISCx78: PostCode
793	SPIx04: SPI_RestrictedCmd	824	MISCx80: StrapStatus
794	SPIx08: SPI_RestrictedCmd2	824	MISCx90: AutoTransaction/Allow EC
794	SPIx0C: SPI_Cntrl1	825	MISCx94: AutoAddrLow
795	SPIx10: SPI_CmdValue0	825	MISCx98: AutoAddrHigh
795	SPIx14: SPI_CmdValue1	825	MISCx9C: AutoData
795	SPIx18: SPI_CmdValue2	825	MISCxC0: CPU Pstate0
796	SPIx1C: Reserved	826	MISCxC4: CPU Pstate1
796	SPIx1D: Alt_SPI_CS	826	MISCxD0: CPU Cstate0
796	SPIx1E: SpiExtRegIndx	826	MISCxD4: CPU Cstate1
796	SPIx1F: SpiExtRegData	827	MISCxF0: SataPortSts
797	SPIx1F_x00: DDR_CMD	827	MISCxF4: ClkCntrlSts
797	SPIx1F_x01: QDR_CMD	827	GPIOx[E4:00]: GPIO
797	SPIx1F_x02: DPR_CMD	828	IOMUXx[E4:00]: IoMux
797	SPIx1F_x03: QPR_CMD	829	PMx00: DecodeEn
797	SPIx1F_x04: ModeByte	829	PMx04: IsaControl
797	SPIx1F_x05: TxByteCount	830	PMx08: PciControl
798	SPIx1F_x06: RxByteCount	831	PMx0C: StpClkSmf
798	SPIx1F_x07: SPIDataFifoPtr	831	PMx20: BiosRamEn
798	SPIx20: SPI100 Enable	832	PMx24: AcpiMmioEn
798	SPIx22: SPI100 Speed Config	832	PMx28: AsfEn
799	SPIx40: DDRCmdCode	833	PMx2C: Smbus0En
799	SPIx41: QDRCmdCode	833	PMx2E: Smbus0Sel
799	SPIx42: DPRCmdCode	833	PMx2F: SmbusSlpGate
799	SPIx43: QPRCmdCode	834	PMx34: IoApicEn
799	SPIx44: ModeByte	834	PMx44: BootTimerEn
800	SPIx45: CmdCode	834	PMx48: WatchdogTimerEn
800	SPIx47: CmdTrigger	835	PMx4C: WatchdogTimerConfig
800	SPIx48: TxByteCount	835	PMx50: HPETEn
800	SPIx4B: RxByteCount	835	PMx54: SerialIrqConfig
800	SPIx4C: SpiStatus	836	PMx56: RTC Control
800	SPIx[C6:80]: FIFO[70:0]	836	PMx58: VRT_T1
801	HPETx000: ID	837	PMx59: VRT_T2
801	HPETx004: ClkPeriod	837	PMx5B: RTC Shadow
801	HPETx010: Config	837	PMx5C: LLBCntrl
801	HPETx020: Interrupt Status	837	PMx5E: RTC ExtIndex
802	HPETx0F0: Main Counter	837	PMx5F: RTC ExtData
802	HPETx1[4:0:Step2]0: Timer[2:0] Config Capability	837	PMx5F_x00: RTCEXT DltSavEnable
802	HPETx1[4:0:Step2]8: Timer[2:0] Comparator	838	PMx5F_x01: RTCEXT SprFwdCtrl
804	MISCx00: GPPClkCntrl	838	PMx5F_x02: RTCEXT SprFwdMonth
806	MISCx04: ClkOutputCntrl	838	PMx5F_x03: RTCEXT FallBackCtrl
808	MISCx08: CGPLLConfig1	838	PMx5F_x04: RTCEXT FallBackMonth
809	MISCx0C: CGPLLConfig2	839	PMx5F_x10: RTCEXT WeekTimerControl

839	PMx5F_x11: RTCEXT WeekTimerReloadLow	861	PM2x00: Fan0InputControl
839	PMx5F_x12: RTCEXT WeekTimerReloadHigh	862	PM2x01: Fan0Control
839	PMx5F_x13: RTCEXT WeekTimerDataLow	862	PM2x02: Fan0Freq
839	PMx5F_x14: RTCEXT WeekTimerDataHigh	863	PM2x03: LowDuty0
840	PMx60: AcpiPm1EvtBlk	863	PM2x04: MedDuty0
840	PMx62: AcpiPm1CntBlk	863	PM2x05: Multiplier0
840	PMx64: AcpiPmTmrBlk	864	PM2x06: LowTemp0Lo
840	PMx66: CpuCntBlk	864	PM2x07: LowTemp0Hi
840	PMx68: AcpiGpe0Blk	864	PM2x08: MedTemp0Lo
840	PMx6A: AcpiSmiCmd	864	PM2x09: MedTemp0Hi
841	PMx6E: AcpiPm2CntBlk	864	PM2x0A: HighTemp0Lo
841	PMx74: AcpiConfig	864	PM2x0B: HighTemp0Hi
841	PMx78: WakeIoAddr	865	PM2x0C: LinearRange0
842	PMx7A: HaltCountEn	865	PM2x0D: LinearHoldCount0
842	PMx7C: CleWrPortAdr	865	PM2x0E: Fan0Hysteresis
842	PMx7E: CStateEn	865	PM2x50: Med2Temp0Lo
842	PMx80: Break Event	865	PM2x51: Med2Temp0Hi
844	PMx88: CStateControl	865	PM2x52: Med2Duty0
844	PMx8E: PopUpEndTime	865	PM2x53: Multiplier2_0
844	PMx94: CStateTiming0	866	PM2x60: FanStatus
844	PMx98: CStateTiming1	866	PM2x61: FanINTRouteLo
845	PMx9C: C2Count	866	PM2x63: SampleFreqDiv
845	PMx9D: C3Count	867	PM2x64: FanDebounceCounterLo
845	PMxA0: MessageCState	867	PM2x65: FanDebounceCounterHi
847	PMxA4: Traffic Monitor and Status	867	PM2x66: Fan0DetectorControl
847	PMxA4_x0: TrafficMonitorEn	867	PM2x67: Fan0SpeedLimitLo
848	PMxA4_x1: TrafficStatus	867	PM2x68: Fan0SpeedLimitHi
849	PMxA8: TrafficMonitorIdleTime	868	PM2x69: Fan0SpeedLo
849	PMxAA: TrafficMonitorIntTime	868	PM2x6A: Fan0SpeedHi
849	PMxAC: TrafficMonitorTrafficCount	868	PM2x8A: TempTsiLo
849	PMxAE: TrafficMonitorIntrCount	868	PM2x8B: TempTsiHi
850	PMxB0: DeferTimeTick	868	PM2x8C: TempTsiLimitLo
850	PMxB4: Tpreset1b	868	PM2x8D: TempTsiLimitHi
851	PMxB8: Tpreset2	868	PM2x8E: TempTsiChangeLimit
851	PMxB9: LpcMisc	869	PM2x8F: TempTsiWe
851	PMxBA: S_StateControl	869	PM2x90: TempTsiStatus
851	PMxBC: ThrottlingControl	869	PM2x92: TempTsiControl
852	PMxBE: ResetControl1	869	PM2x94: TempTsiINTRoute
853	PMxC0: S5/Reset Status	870	PM2xDF: TempTsiRstSel
854	PMxC4: ResetCommand	870	PM2xE0: AlertThermaltripStatus
854	PMxC5: CF9 Shadow	870	PM2xE1: AlertLimitLo
854	PMxC8: Misc	870	PM2xE2: AlertLimitHi
855	PMxCC: IoDrvSth	870	PM2xE3: ThermalTripLimitLo
856	PMxD0: RstCntrl	871	PM2xE4: ThermalTripLimitHi
856	PMxD1: Reset Function	871	PM2xE5: AlertThermaltripControl
856	PMxD1_x0: RstLength	871	AcpiPmEvtBlkx00: Pm1Status
856	PMxD1_x1: APURstLength	872	AcpiPmEvtBlkx02: Pm1Enable
856	PMxD1_x3: APUPwrGdLength	872	AcpiPm1CntBlkx00: PmControl
857	PMxD2: PmioDebug	873	AcpiPm2CntBlkx00: Pm2Control
857	PMxD3: ManualReset	873	AcpiPmTmrBlkx00: TmrValue/ETmrValue
857	PMxD6: IMC Gating	873	CpuCntBlkx00: ClkValue
857	PMxD8: Eprom Index	874	CpuCntBlkx04: PLv12
857	PMxD9: Eprom Data	874	CpuCntBlkx05: PLv13
858	PMxDA: SataConfig	874	AcpiGpe0Blkx00: EventStatus
858	PMxDC: SataConfig2	874	AcpiGpe0Blkx04: EventEnable
858	PMxE0: ABRegBar	874	SmiCmdBlkx00: SmiCmdPort
859	PMxE4: ABDebug	874	SmiCmdBlkx01: SmiCmdStatus
859	PMxE8: SDFlashCntrl	875	SMIx00: Event_Status
859	PMxEB: AzEn	875	SMIx04: Event_Enable
859	PMxEC: LpcGating	875	SMIx08: SciTrig
860	PMxED: USB Gating	877	SMIx0C: SciLevl
860	PMxEF: USB Enable	878	SMIx10: SmiSciStatus
860	PMxF0: USB Control	878	SMIx14: SmiSciEn



878	SMIx18: SwSciEn	905	AcDcTimerx10: DcTimerValue
878	SMIx1C: SwSciData	905	AcDcTimerx14: DcExpiredTimerPolicy
878	SMIx20: SciSleepDisable	905	AcDcTimerx18: DcTimerStatus
879	SMIx30: CapturedData	905	AcDcTimerx20: AcDcTimerCtrl
879	SMIx34: CapturedValid		
879	SMIx38: EPBIF_AER_Straps		
880	SMIx3C: DataErrorStatus		
880	SMIx40: SciMap0		
881	SMIx44: SciMap1		
881	SMIx48: SciMap2		
882	SMIx4C: SciMap3		
882	SMIx50: SciMap4		
883	SMIx54: SciMap5		
883	SMIx58: SciMap6		
884	SMIx5C: SciMap7		
884	SMIx60: SciMap8		
884	SMIx64: SciMap9		
884	SMIx68: SciMap10		
885	SMIx6C: SciMap11		
885	SMIx70: SciMap12		
886	SMIx74: SciMap13		
886	SMIx78: SciMap14		
887	SMIx7C: SciMap15		
887	SMIx80: SmiStatus0		
888	SMIx84: SmiStatus1		
889	SMIx88: SmiStatus2		
890	SMIx8C: SmiStatus3		
890	SMIx90: SmiStatus4		
891	SMIx94: SmiPointer		
892	SMIx96: SmiTimer		
892	SMIx98: SmiTrig0		
893	SMIx9C: SmiTrig1		
893	SMIxA0: SmiControl0		
894	SMIxA4: SmiControl1		
894	SMIxA8: SmiControl2		
895	SMIxAC: SmiControl3		
896	SMIxB0: SmiControl4		
896	SMIxB4: SmiControl5		
897	SMIxB8: SmiControl6		
897	SMIxBC: SmiControl7		
898	SMIxC0: SmiControl8		
898	SMIxC4: SmiControl9		
899	SMIx8: IoTrapping0		
899	SMIxCA: IoTrapping1		
899	SMIxCC: IoTrapping2		
900	SMIxCE: IoTrapping3		
900	SMIx D0: MemTrapping0		
900	SMIx D4: MemRdOvrData0		
900	SMIx D8: MemTrapping1		
900	SMIx DC: MemRdOvrData1		
900	SMIx E0: MemTrapping2		
901	SMIx E4: MemRdOvrData2		
901	SMIx E8: MemTrapping3		
901	SMIx EC: MemRdOvrData3		
901	SMIx F0: CfgTrapping0		
901	SMIx F4: CfgTrapping1		
902	SMIx F8: CfgTrapping2		
902	SMIx FC: CfgTrapping3		
903	WDTx00: WatchdogControl		
903	WDTx04: WatchdogCount		
904	AcDcTimerx00: AcTimerValue		
904	AcDcTimerx04: AcExpiredTimerPolicy		
904	AcDcTimerx08: AcTimerStatus		